Clocked SR Latch

All of the SR latch circuits examined in the previous section are essentially asynchronous sequential circuits, which will respond to the changes occurring in input signals at a circuit-delay-dependent time point during their operation. To facilitate synchronous operation, the circuit response can be controlled simply by adding a gating clock signal to the circuit, so that the outputs will respond to the input levels only during the active period of a clock pulse. For simple reference, the clock pulse will be assumed to be a periodic square waveform, which is applied simultaneously to all clocked logic gates in the system.

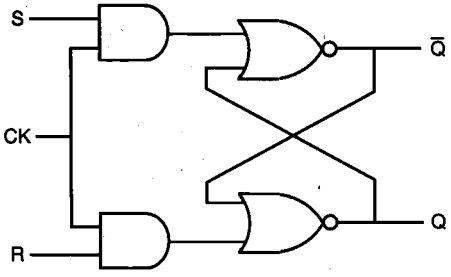


Figure 8.14. Gate-level schematic of the clocked NOR-based SR latch.

The gate-level schematic of a clocked NOR-based SR latch is shown in Fig. 8.14. It can be seen that if the clock (CK) is equal to logic "0," the input signals have no influence upon the circuit response. The outputs of the two AND gates will remain at logic "0," which forces the SR latch to hold its current state regardless of the S and R input signals. When the clock input goes to logic "1," the logic levels applied to the S and R inputs are permitted to reach the SR latch, and possibly change its state. Note that as in the non-clocked SR latch, the input combination S = R = "1" is not allowed in the clocked SR latch. With both inputs S and R at logic "1," the occurrence of a clock pulse causes both outputs to go momentarily to zero. When the clock pulse is removed, i.e., when it becomes "0," the state of the latch is indeterminate. It can eventually settle into either state, depending on slight delay differences between the output signals.

To illustrate the operation of the clocked SR latch, a sample sequence of CK, S, and R waveforms, and the corresponding output waveform Q are shown in Fig. 8.15. Note that the circuit is strictly *level-sensitive* during active clock phases, i.e., any changes occurring in the S and R input voltages when the CK level is equal to "1" will be reflected onto the circuit outputs. Consequently, even a narrow spike or glitch occurring during an active clock phase can set or reset the latch, if the loop delay is shorter than the pulse width.

Figure 8.16 shows a CMOS implementation of the clocked NOR-based SR latch circuit, using two simple AOI gates. Notice that the AOI-based implementation of the circuit results in a very small transistor count, compared with the alternative circuit realization consisting of two AND2 and two NOR2 gates.

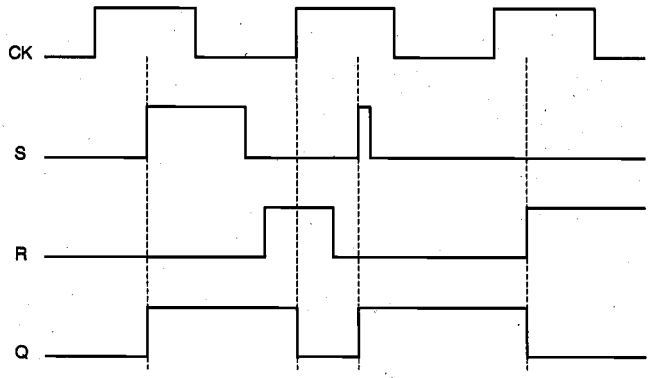


Figure 8.15. Sample input and output waveforms illustrating the operation of the clocked NOR-based SR latch circuit.

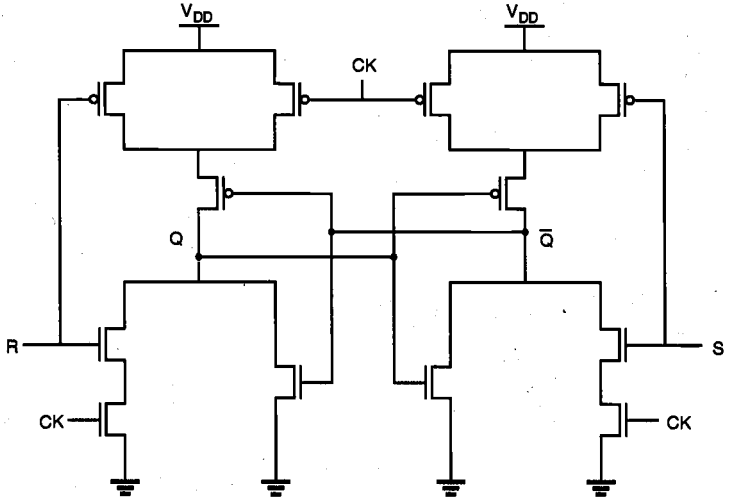


Figure 8.16. AOI-based implementation of the clocked NOR-based SR latch circuit.

The NAND-based SR latch can also be implemented with gating clock input, as shown in Fig. 8.17. It must be noted, however, that both input signals S and R as well as the clock signal CK are *active low* in this case. This means that changes in the input signal levels will be ignored when the clock is equal to logic "1," and that inputs will influence the outputs only when the clock is active, i.e., CK = "0." For the circuit implementation of this clocked NAND-based SR latch, we can use a simple OAI structure, which is essentially analogous to the AOI-based realization of the clocked NOR SR latch circuit.

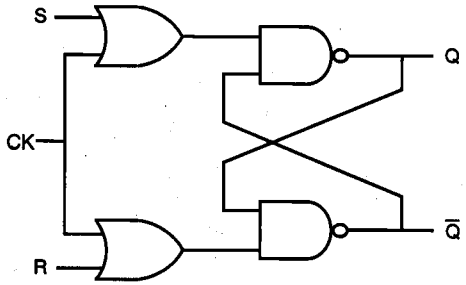
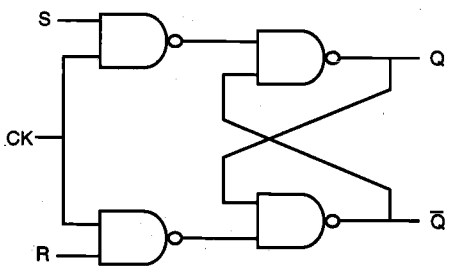
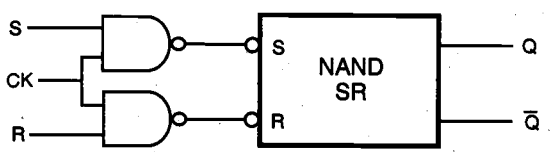


Figure 8.17. Gate-level schematic of the clocked NAND-based SR latch circuit, with active low inputs.

A different implementation of the clocked NAND-based SR latch is shown in Fig. 8.18. Here, both input signals and the CK signal are *active high*, i.e., the latch output Q will be set when CK = "1," S = "1," and R = "0." Similarly, the latch will be reset when CK = "1," S = "0," and R = "1." The latch preserves its state as long as the clock signal is inactive, i.e., when CK = "0." The drawback of this implementation is that the transistor count is higher than the *active low* version shown in Fig. 8.17.



(a)



(b)

Figure 8.18. (a) Gate-level schematic of the clocked NAND-based SR latch circuit, with active high inputs. (b) Partial block diagram representation of the same circuit.

All simple and clocked SR latch circuits examined to this point suffer from the common problem of having a not-allowed input combination, i.e., their state becomes indeterminate when both inputs S and R are activated at the same time. This problem can be overcome by adding two feedback lines from the outputs to the inputs, as shown in Fig. 8.19. The resulting circuit is called a JK latch. Figure 8.19 shows an all-NAND implementation of the JK latch with active high inputs, and the corresponding block diagram representation. The JK latch is commonly called a JK flip-flop.

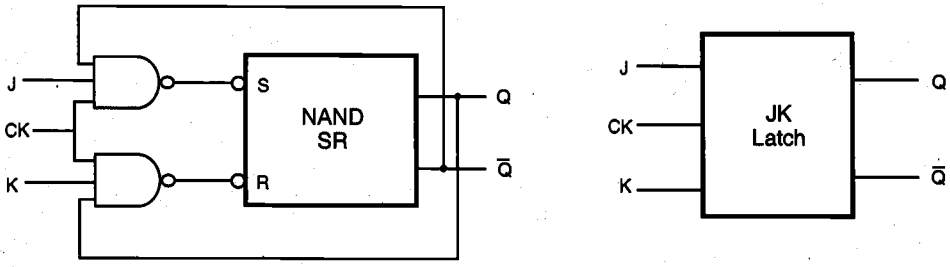


Figure 8.19. Gate-level schematic of the clocked NAND-based JK latch circuit.

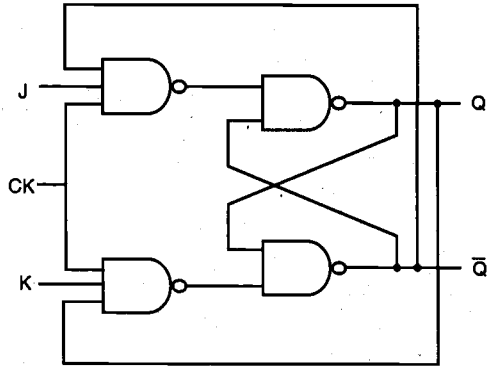


Figure 8.20. All-NAND implementation of the clocked JK latch circuit.

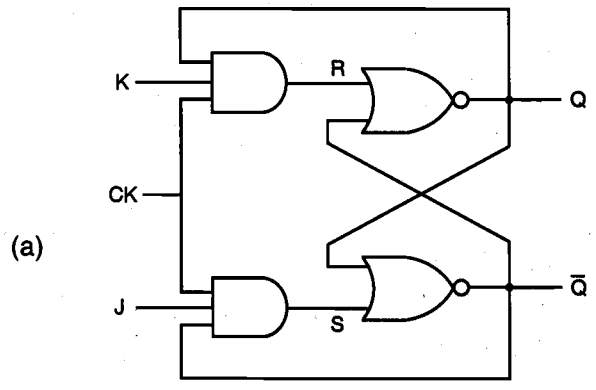
The J and K inputs in this circuit correspond to the set and reset inputs of the basic SR latch. When the clock is active, the latch can be set with the input combination (J = "1," K = "0"), and it can be reset with the input combination (J = "0," K = "1"). If both inputs are equal to logic "0," the latch preserves its current state. If, on the other hand, both inputs are equal to "1" during the active clock phase, the latch simply switches its state due to feedback. In other words, the JK latch does not have a not-allowed input combination. As in the other clocked latch circuits, the JK latch will hold its current state when the clock is inactive (CK = "0"). The operation of the clocked JK latch is summarized in the truth table (Table 8.3).

Figure 8.21 shows an alternative, NOR-based implementation of the clocked JK latch, and CMOS realization of this circuit. Note that the AOI-based circuit structure results in a relatively low transistor count, and consequently, a more compact circuit compared to the all-NAND realization shown in Fig. 8.20.

J	K	Q_n	$\overline{Q_n}$	S	R	Q_{n+1}	$\overline{Q_{n+1}}$	*Operation*
0	0	0	1	1	1	0	1	*hold*
		1	0	1	1	1	0	
0	1	0	1	1	1	0	1	*reset*
		1	0	1	0	0	1	
1	0	0	1	0	1	1	0	*set*
		1	0	1	1	1	0	
1	1	0	1	0	1	1	0	*toggle*
		1	0	1	0	0	1	

Table 8.3. Detailed truth table of the JK latch circuit.

While there is no not-allowed input combination for the JK latch, there is still a potential problem. If both inputs are equal to logic "1" during the active phase of the clock pulse, the output of the circuit will oscillate (toggle) continuously until either the clock becomes inactive (goes to zero), or one of the input signals goes to zero. To prevent this undesirable timing problem, the clock pulse width must be made smaller than the



(a)

Figure 8.21. (a) Gate-level schematic of the clocked NOR-based JK latch circuit.

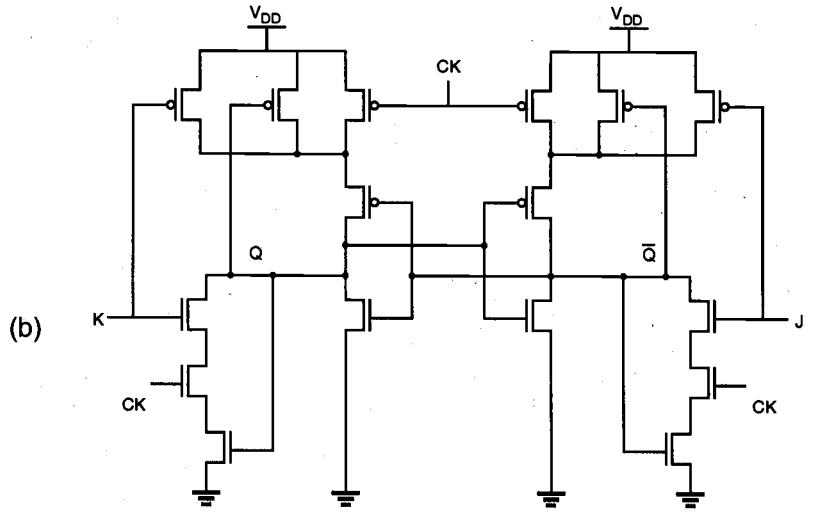


Figure 8.21. (continued) (b) CMOS AOI realization of the JK latch.

input-to-output propagation delay of the JK latch circuit. This restriction dictates that the clock signal must go low before the output level has an opportunity to switch again, which prevents uncontrolled oscillation of the output. However, note that this clock constraint is difficult to implement for most practical applications.

Assuming that the clock timing constraint described above is satisfied, the output of the JK latch will toggle (change its state) only once for each clock pulse, if both inputs are equal to logic "1" (Fig. 8.22). A circuit which is operated exclusively in this mode is called a *toggle switch*.

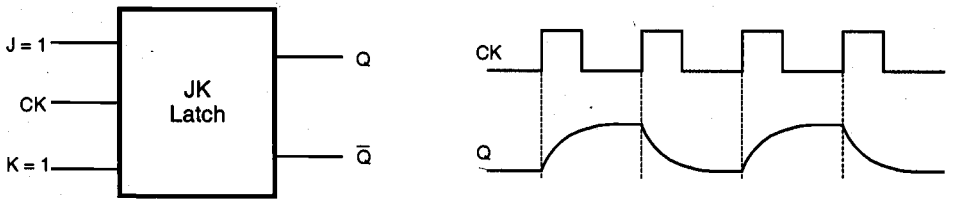


Figure 8.22. Operation of the JK latch as a toggle switch.

Master-Slave Flip-Flop

Most of the timing limitations encountered in the previously examined clocked latch circuits can be prevented by using two latch stages in a cascaded configuration. The key

operation principle is that the two cascaded stages are activated with opposite clock phases. This configuration is called the *master-slave flip-flop*. Our definition of flip-flop is designed to distinguish it from latches discussed previously, although they are mostly used interchangeably in the literature.

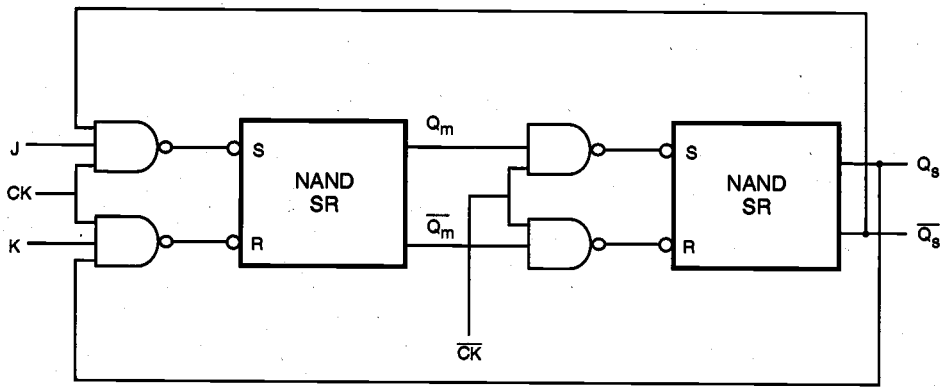


Figure 8.23. Master-slave flip-flop consisting of NAND-based JK latches.

The input latch in Fig. 8.23, called the "master," is activated when the clock pulse is high. During this phase, the inputs J and K allow data to be entered into the flip-flop, and the first-stage outputs are set according to the primary inputs. When the clock pulse goes to zero, the master latch becomes inactive and the second-stage latch, called the "slave," becomes active. The output levels of the flip-flop circuit are determined during this second phase, based on the master-stage outputs set in the previous phase.

Since the master and the slave stages are effectively decoupled from each other with the opposite clocking scheme, the circuit is never *transparent*, i.e., a change occurring in the primary inputs is never reflected directly to the outputs. This very important property clearly separates the master-slave flip-flop from all of the latch circuits examined earlier in this section. Figure 8.24 shows a sample set of input and output waveforms associated with the JK master-slave flip-flop, which can help the reader to study the basic operation principles.

Because the master and the slave stages are decoupled from each other, the circuit allows for toggling when J = K = "1," but it eliminates the possibility of uncontrolled oscillations since only one stage is active at any given time. A NOR-based alternative realization for the master-slave flip-flop circuit is shown in Fig. 8.25.

Figure 8.24 also shows that the master-slave flip-flop circuit examined here has the potential problem of "one's catching." When the clock pulse is high, a narrow spike or glitch in one of the inputs, for instance a glitch in the J line (or K line), may set (or reset) the master latch and thus cause an unwanted state transition, which will then be propagated into the slave stage during the following phase. This problem can be eliminated to a large extent by building an edge-triggered master-slave flip-flop, which will be examined in the following section.

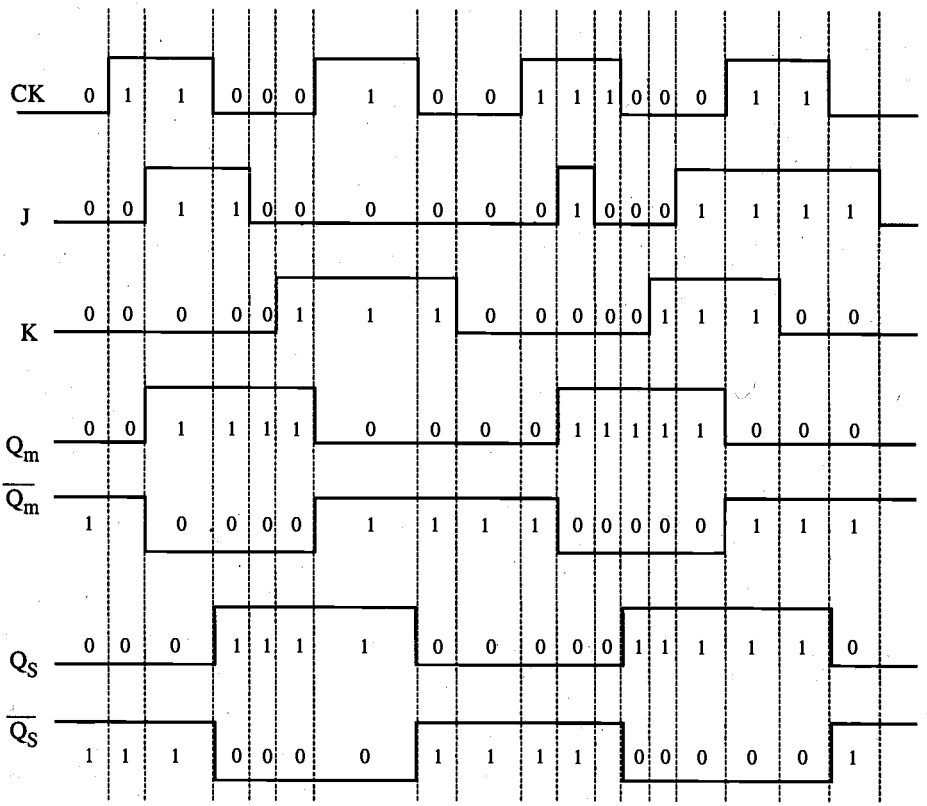


Figure 8.24. Sample input and output waveforms of the master-slave flip-flop circuit.

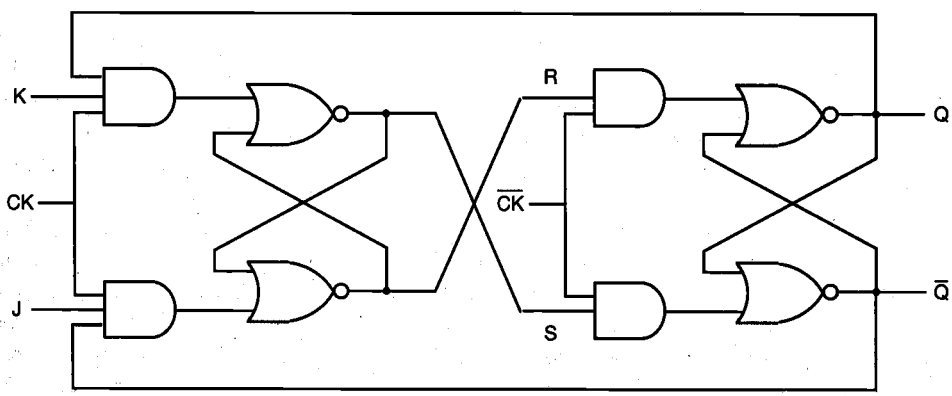


Figure 8.25. NOR-based realization of the JK master-slave flip-flop.

8.5. CMOS D-Latch and Edge-Triggered Flip-Flop

With the widespread use of CMOS circuit techniques in digital integrated circuit design, a large selection of CMOS-based sequential circuits have also gained popularity and prominence, especially in VLSI design. Throughout this chapter, we have seen examples showing that virtually all of the latch and flip-flop circuits can be implemented with CMOS gates, and that their design is quite straightforward. However, direct CMOS implementations of conventional circuits such as the clocked JK latch or the JK master-slave flip-flop tend to require a large number of transistors.

In this section, we will see that specific versions of sequential circuits built primarily with CMOS transmission gates are generally simpler and require fewer transistors than the circuits designed with conventional structuring. As an introduction to the issue, let us first consider the simple D-latch circuit shown in Fig. 8.26. The gate-level representation of the D-latch is simply obtained by modifying the clocked NOR-based SR latch circuit. Here, the circuit has a single input D, which is directly connected to the S input of the latch. The input variable D is also inverted and connected to the R input of the latch. It can be seen from the gate-level schematic that the output Q assumes the value of the input D when the clock is active, i.e., for CK = "1." When the clock signal goes to zero, the output will simply preserve its state. Thus, the CK input acts as an enable signal which allows data to be accepted into the D-latch.

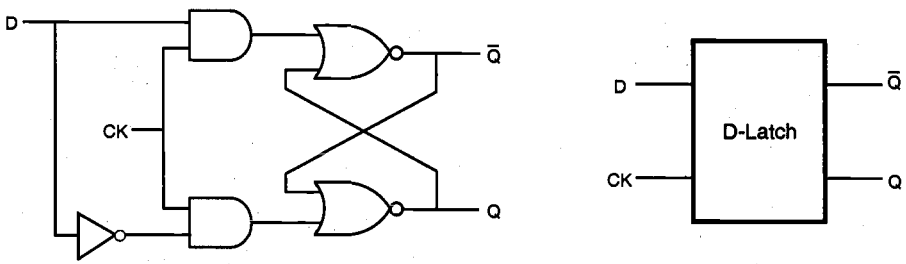


Figure 8.26. Gate-level schematic and the block diagram view of the D-latch.

The D-latch finds many applications in digital circuit design, primarily for temporary storage of data or as a delay element. In the following, we will examine its simple CMOS implementation. Consider the circuit diagram given in Fig. 8.27, which shows a basic two-inverter loop and two CMOS transmission gate (TG) switches.

The TG at the input is activated by the CK signal, whereas the TG in the inverter loop is activated by the inverse of the CK signal, $\overline{CK}$. Thus, the input signal is accepted (latched) into the circuit when the clock is high, and this information is preserved as the state of the inverter loop when the clock is low. The operation of the CMOS D-latch circuit can be better visualized by replacing the CMOS transmission gates with simple switches, as shown in Fig. 8.28. A timing diagram accompanying this figure shows the time intervals during which the input and the output signals should be valid (unshaded).

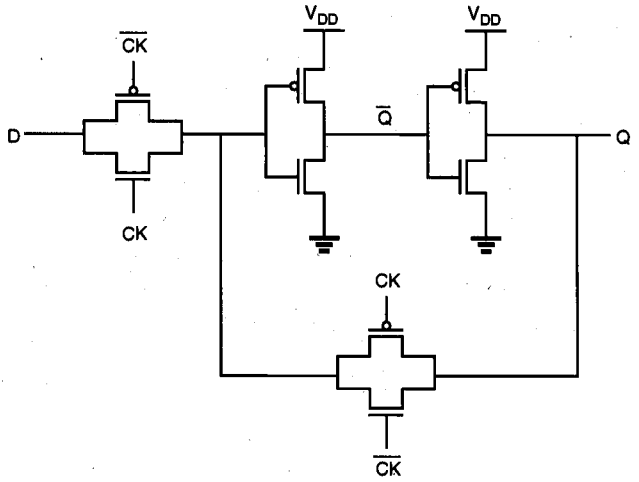


Figure 8.27. CMOS implementation of the D-latch (version 1).

Note that the valid D input must be stable for a short time before (*setup time*, t_{setup}) and after (*hold time*, t_{hold}) the negative clock transition, during which the input switch opens and the loop switch closes. Once the inverter loop is completed by closing the loop switch, the output will preserve its valid level. In the D-latch design, the requirements for setup time and hold time should be met carefully. Any violation of such specifications can cause *metastability* problems which lead to seemingly chaotic transient behavior, and can result in an unpredictable state after the transitional period.

The D-latch shown in Fig. 8.27 is not an edge-triggered storage element because the output changes according to the input, i.e., the latch is transparent, while the clock is high. The transparency property makes the application of this D-latch unsuitable for counters and some data storage implementations.

Figure 8.29 shows a different version of the CMOS D-latch. The circuit contains two tristate inverters, driven by the clock signal and its inverse. Although the circuit appears to be quite different from that shown in Fig. 8.27, the basic operation principle of the circuit is the same as that shown in Fig. 8.28. The first tri-state inverter acts as the input switch, accepting the input signal when the clock is high. At this time, the second tristate inverter is at its high-impedance state, and the output Q is following the input signal. When the clock goes low, the input buffer becomes inactive, and the second tristate inverter completes the two-inverter loop, which preserves its state until the next clock pulse.

Finally, consider the two-stage master-slave flip-flop circuit shown in Fig. 8.30, which is constructed by simply cascading two D-latch circuits. The first stage (master) is driven by the clock signal, while the second stage (slave) is driven by the inverted clock signal. Thus, the master stage is positive level-sensitive, while the slave stage is negative level-sensitive.

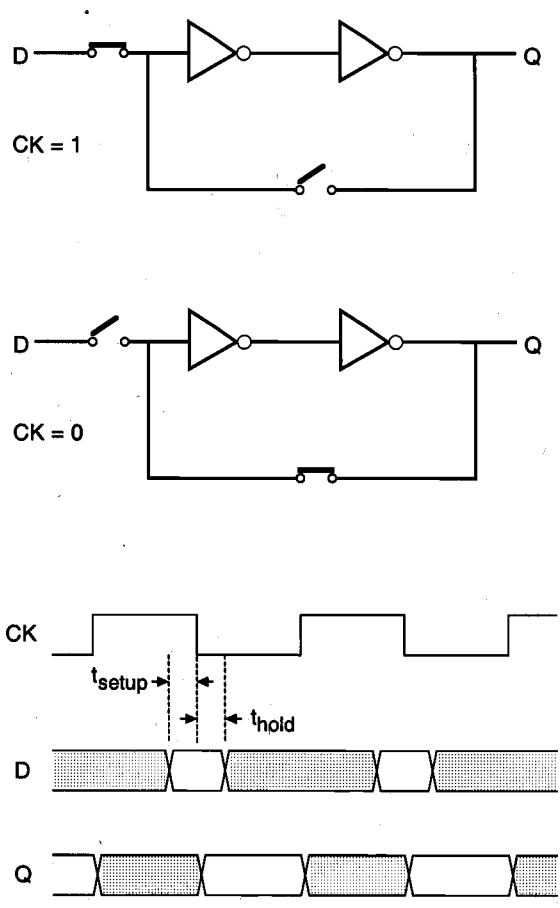


Figure 8.28. Simplified schematic view and the corresponding timing diagram of the CMOS D-latch circuit, showing the setup time and the hold time.

When the clock is high, the master stage follows the D input while the slave stage holds the previous value. When the clock changes from logic "1" to logic "0," the master latch ceases to sample the input and stores the D value at the time of the clock transition. At the same time, the slave latch becomes transparent, passing the stored master value Q_m to the output of the slave stage, Q_s. The input cannot affect the output because the master stage is disconnected from the D input. When the clock changes again from logic "0" to "1," the slave latch locks in the master latch output and the master stage starts sampling the input again. Thus, this circuit is a negative edge-triggered D flip-flop by virtue of the fact that it samples the input at the falling edge of the clock pulse.

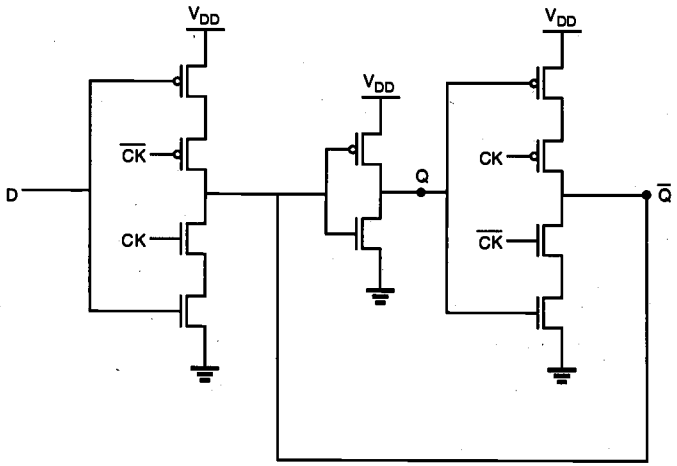


Figure 8.29. CMOS implementation of the D-latch (version 2).

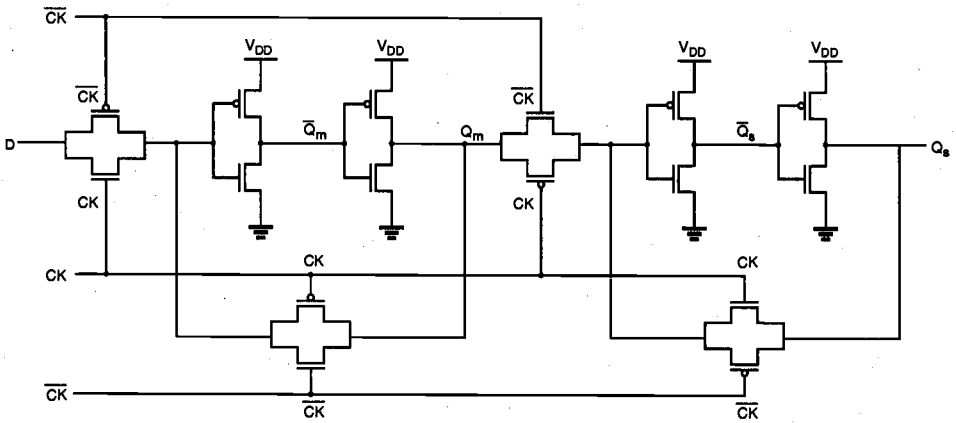


Figure 8.30. CMOS negative (falling) edge-triggered master-slave D flip-flop (DFF).

Figure 8.31 shows the simulated input and output waveforms of the CMOS negative edge-triggered D-type flip-flop. The output of the master stage latches the applied input (D) when the clock signal is "1", and the output of the slave stage becomes valid when the clock signal drops to "0". Thus, the D-type flip-flop (DFF) essentially samples the input at every falling edge of the clock pulse.

It should be emphasized that the operation of the DFF circuit can be seriously affected if the master stage experiences a set-up time violation. This situation is illustrated in Fig. 8.32, where the input D switches from "0" to "1" immediately before

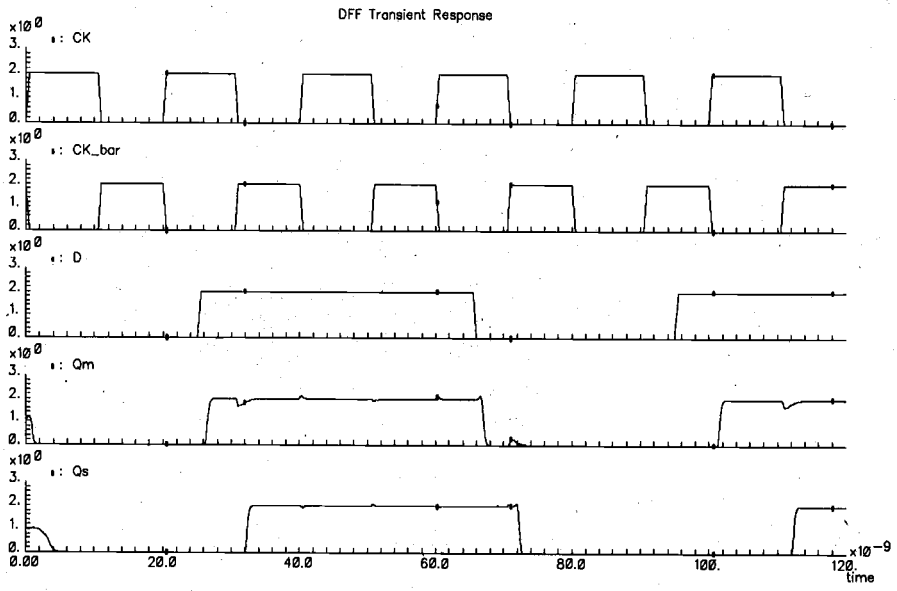


Figure 8.31. Simulated input and output waveforms of the CMOS DFF circuit in Fig. 8.30.

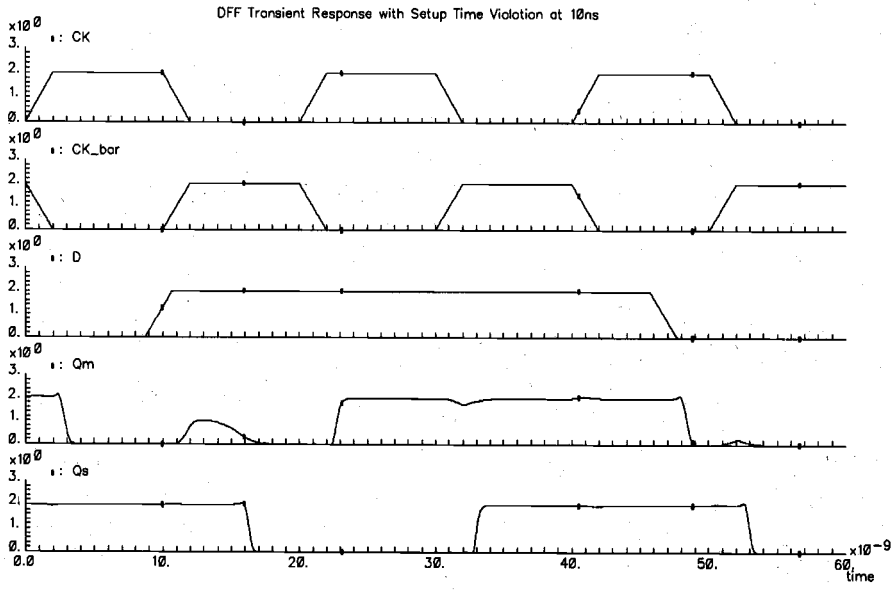


Figure 8.32. Simulated waveforms of the CMOS DFF circuit, showing a set-up time violation for the master stage input at 10 ns. The output of the master stage fails to settle at the correct level.

the clock transition occurs (set-up time violation). As a result, the master stage fails to latch the correct value, and the slave stage produces an erroneous output. The relative timing of the input and clock signals are carefully synchronized to avoid such situations. The layout of the CMOS DFF circuit is given in Fig. 8.33.

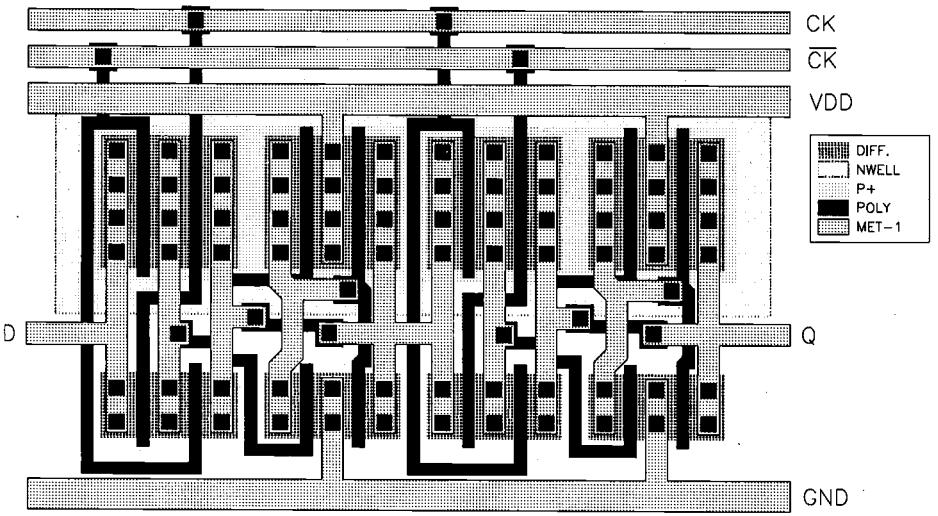


Figure 8.33. Layout of the CMOS DFF shown in Fig. 8.30.

Another implementation of edge-triggered D flip-flop is shown in Fig. 8.34, which consists of six NAND3 gates. This D flip-flop is positive edge-triggered as illustrated in the waveform chart in Fig. 8.35. Initially, all the signal values except for S are 0, i.e., (S, R, CK, D) = (1, 0, 0, 0), and Q = 0. In the second phase, both D and R switch to 1, i.e., (S, R, CK, D) = (1, 0, 1, 1), but no change in Q occurs and the Q value remains at 0. However, in the third phase, if CK goes to high, i.e., (S, R, CK, D) = (1, 1, 1, 1), the output of gate 2 switches to 0, which in turn sets the output of the last stage SR latch to 1. Thus, the output of this D flip-flop switches to 1 at the positive-going edge of the clock signal, CK. However, as can be observed in the ninth phase of the waveform diagram chart, the Q output is not affected by the negative-going edge of CK, nor by other signal changes.

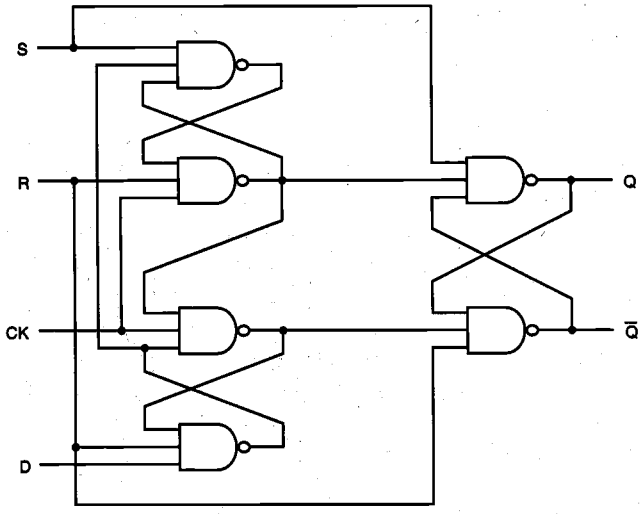


Figure 8.34. NAND3-based positive edge-triggered D flip-flop circuit.

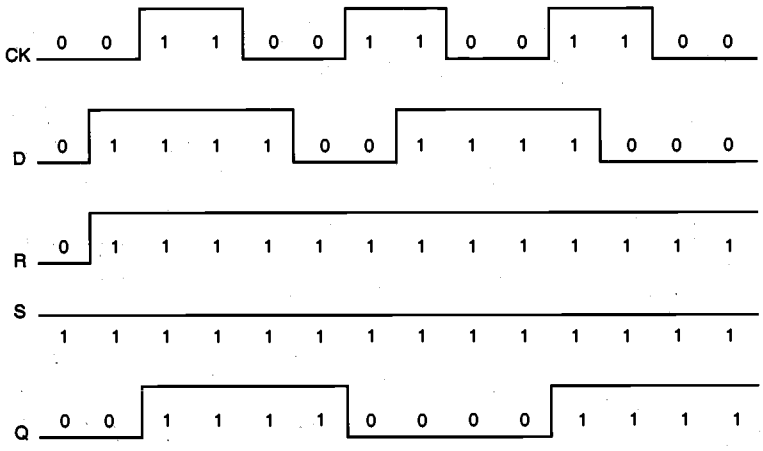


Figure 8.35. Timing diagram of the positive edge-triggered D flip-flop.