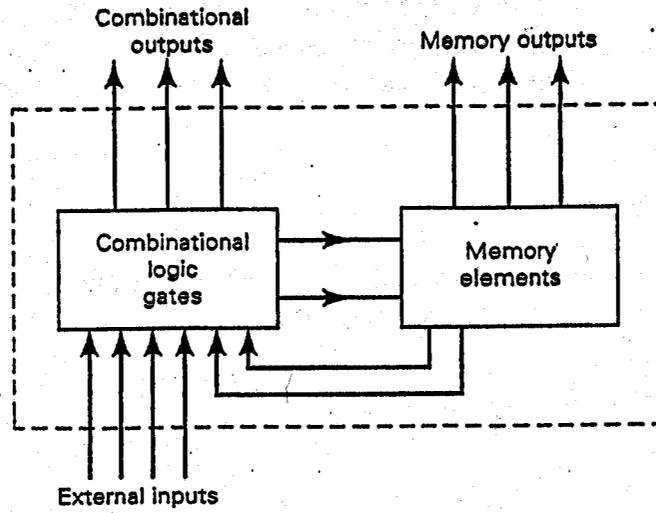


FLIP FLOPS

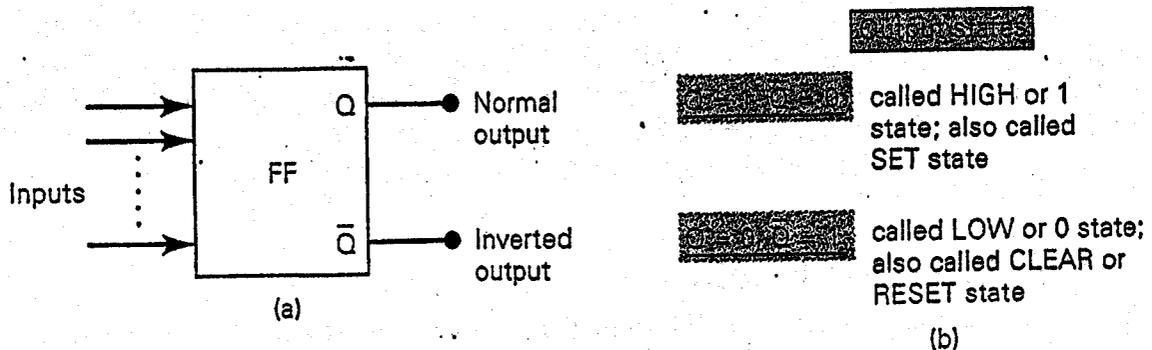
INTRODUCTION

The logic circuits considered thus far have been combinational circuits whose output levels at any instant of time are dependent on the levels present at the inputs at that time. Any prior input-level conditions have no effect on the present outputs because combinational logic circuits have no memory. Most digital systems are made up of both combinational circuits and memory elements.



The combinational portion accepts logic signals from external inputs and from the outputs of the memory elements. The combinational circuit operates on these inputs to produce various outputs, some of which are used to determine the binary values to be stored in the memory elements. The outputs of some of the memory elements, in turn, go to the inputs of logic gates in the combinational circuits. This process indicates that the external outputs of a digital system are a function of both its external inputs and the information stored in its memory elements.

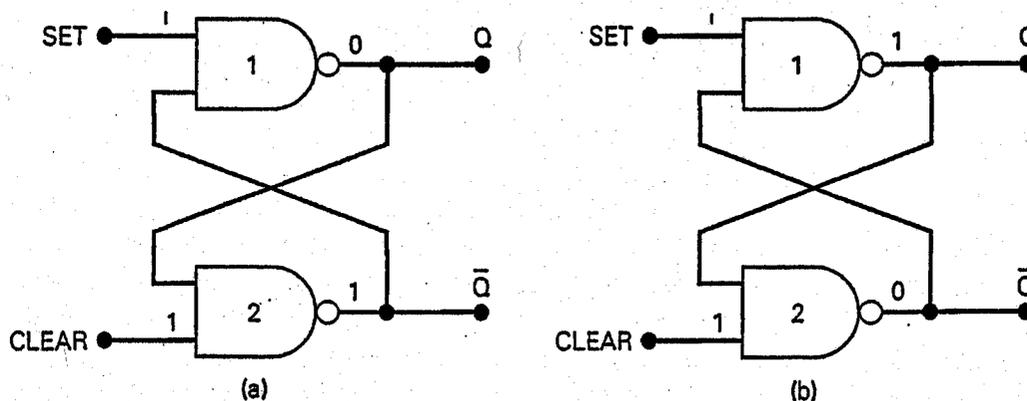
The most important memory element is the flip-flop, which is made up of an assembly of logic gates. Even though a logic gate, by itself, has no storage capability, several can be connected together in ways that permit information to be stored. Several different gate arrangements are used to produce these flip-flops (FF).



The two outputs are the inverse of each other. The Q output is called the *normal FF* output, and the other is the *inverted FF* output. Note that the HIGH or 1 state is also referred to as the SET state. Whenever the inputs to a FF cause it to go to the $Q = 1$ state, we call this *setting the FF*; the FF has been set. Whenever the inputs to a FF cause it to go to the $Q = 0$ state, we call this *clearing* or *resetting* the FF; the FF has been cleared (reset). The inputs are used to cause the FF to switch back and forth ("flip-flop") between its possible output states. Most FF inputs need only to be momentarily activated (pulsed) in order to cause a change in the FF output state, and the output will remain in that new state even after the input pulse is over. This is the FF's *memory* characteristic. The flip-flop is known by other names, including *latch* and *bistable multivibrator*

NAND GATE LATCH

The most basic FF circuit can be constructed from either two NAND gates or two NOR gates. The NAND gate version is called a NAND gate latch.



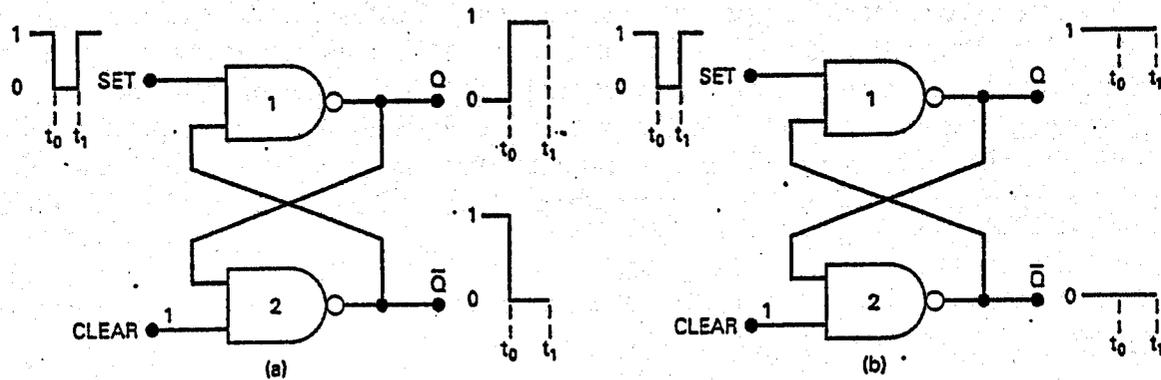
The two NAND gates are cross-coupled so that the output of NAND-1 is connected to one of the inputs of NAND-2, and vice versa. The gate outputs are the latch outputs. Under normal conditions, these outputs will always be the inverse of each other. There are two latch inputs: the SET input is the input that *sets* Q to the 1 state; the CLEAR input is the input that *clears* Q to the 0 state. The SET and CLEAR inputs are both normally resting in the HIGH state, and one of them will be pulsed LOW whenever we want to change the latch outputs.

There are two equally likely output states when $SET = CLEAR = 1$; one possibility is shown in (a). In effect, the LOW at the NAND-1 output produces a HIGH at the NAND-2 output, which, in turn, keeps the NAND-1 output LOW. The second possibility is shown in (b), where the HIGH from NAND-1 produces a LOW at the NAND-2 output, which, in turn, keeps the NAND-1 output HIGH. Thus, there are two possible output states when $SET = CLEAR = 1$; the one that actually exists will depend on what has occurred previously at the inputs.

SETTING THE LATCH (FF)

Consider $Q=0$ initially. When the SET input is momentarily pulsed LOW while CLEAR is kept HIGH, Q will go HIGH, and this HIGH will force the NAND-2 output to go LOW so that

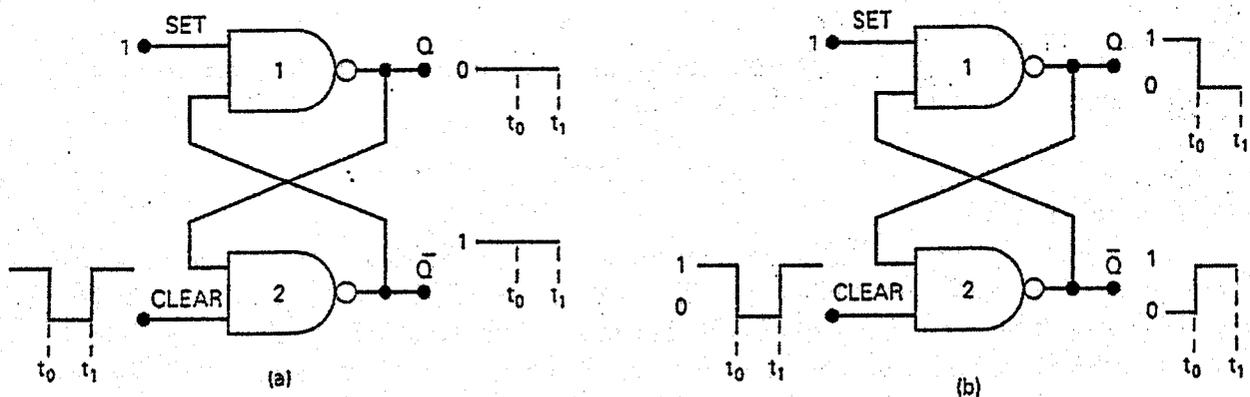
NAND-1 now has two LOW inputs. Thus, when SET returns to the 1 state after the pulse, the NAND-1 output *remains* HIGH, which, in turn, keeps the NAND-2 output LOW.



(b) illustrates what happens when $Q = 1$ prior to the application of the SET pulse. The LOW pulse at SET will not change anything. Thus, when SET returns HIGH, the latch outputs are still in the $Q = 1$ state. To summarize, a LOW pulse on the SET input will always cause the latch to end up in the $Q = 1$ state. This operation is called *setting* the latch or FF.

CLEARING THE LATCH (FF)

Consider $Q=0$ initially. When the CLEAR input is pulsed LOW while SET is kept HIGH, the LOW pulse at CLEAR will not have any effect, since $Q = 0$ is already keeping the NAND-2 output *HIGH*.



When CLEAR returns HIGH, the latch outputs still have $Q = 0$. (b) shows the situation when $Q = 1$ prior to the occurrence of the CLEAR pulse. As CLEAR is pulsed LOW, the NAND-2 output will go HIGH, and this HIGH forces Q to go LOW, so that NAND-2 now has two LOW inputs. Thus, when the CLEAR pulse returns HIGH, the NAND-2 output *remains* HIGH, which, in turn, keeps the NAND-1 output LOW. To summarize, a LOW pulse on the CLEAR input will always cause the latch to end up in the $Q = 0$ state. This operation is called *clearing* or *resetting* the latch.

SIMULTANEOUS SETTING AND CLEARING

The last case to consider is the case where the SET and CLEAR inputs are simultaneously pulsed LOW. This will produce HIGH levels at both NAND outputs. Clearly, this is an undesired condition, since the two outputs are supposed to be inverses of each other. Furthermore, when the SET and CLEAR inputs return HIGH, the resulting output state will depend on which input returns HIGH first. Simultaneous transitions back to the 1 state will produce unpredictable results. For these reasons the SET = CLEAR = 0 condition is normally not used for the NAND latch.

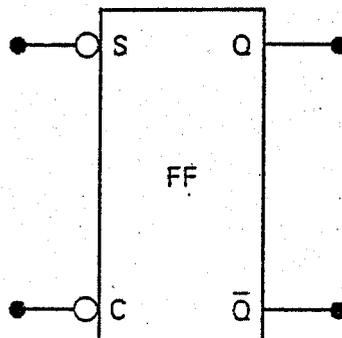
SUMMARY OF THE NAND LATCH

The operation described above can be conveniently placed in a truth table and as follows:

1. SET = CLEAR = 1. This condition is the normal resting state, and it has no effect on the output state. The Q outputs will remain in whatever state they were in prior to this input condition.
2. SET = 0, CLEAR = 1. This will always cause the output to go to the Q = 1 state, where it will remain even after SET returns HIGH. This is called *setting* the latch.
3. SET = 1, CLEAR = 0. This will always produce the Q = 0 state, where the output will remain even after CLEAR returns HIGH. This is called *clearing* or *resetting* the latch.
4. SET = CLEAR = 0. This condition tries to set and clear the latch at the same time and can produce ambiguous results. It should not be used.

REPRESENTATION

From the description of the NAND latch operation, it should be clear that the SET and CLEAR inputs are active-LOW. The SET input will set Q = 1 when SET goes LOW; the CLEAR input will clear Q = 0 when CLEAR goes LOW. For this reason, the NAND latch is often drawn using the representation shown below. The S and C labels represent the SET and CLEAR inputs, and the bubbles indicate the active-LOW nature of these inputs.



TERMINOLOGY

The action of *clearing* a FF or a latch is also called *resetting*, and both terms are used interchangeably in the digital field. A CLEAR input can also be called a RESET input, and a SET-CLEAR latch can be called a SET-RESET (SR) latch.

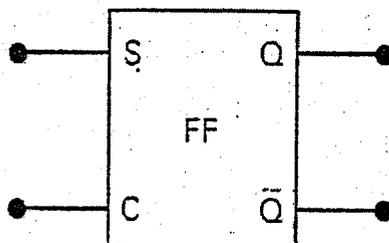
It is virtually impossible to obtain a "clean" voltage transition from a mechanical switch, because of the phenomenon of **contact bounce**, where the action of moving the switch produces several output voltage transitions as the switch bounces (makes and breaks contact) before coming to rest. The multiple transitions on the output signal generally last no longer than a few milliseconds, but they would be unacceptable in many applications. A NAND latch can be used to prevent the presence of switch bounce from affecting the output.

NOR GATE LATCH

The analysis and operation of the NOR latch can be performed in exactly the same manner as for the NAND latch. The results are as follows:

1. **SET = CLEAR = 0.** This is the normal resting state for the NOR latch, and it has no effect on the output state. The outputs will remain in whatever state they were in prior to the occurrence of this input condition.
2. **SET = 1, CLEAR = 0.** This will always set $Q = 1$, where it will remain even after SET returns to 0.
3. **SET = 0, CLEAR = 1.** This will always clear $Q = 0$, where it will remain even after CLEAR returns to 0.
4. **SET = 1, CLEAR = 1.** This condition tries to set and clear the latch at the same time. If the inputs are returned to 0 simultaneously, the resulting output state is unpredictable. This input condition should not be used.

The NOR gate latch operates exactly like the NAND latch except that the SET and CLEAR inputs are active-HIGH rather than active-LOW, and the normal resting state is SET = CLEAR = 0. Q will be set HIGH by a HIGH pulse on the SET input, and it will be cleared LOW by a HIGH pulse on the CLEAR input. The simplified block symbol for the NOR shows no bubbles on the S and C inputs; this indicates that these inputs are active-HIGH.



(c)