Joseph J. Carr K4IPV
5440 South 8th Road
Arlington VA 22204

Digital Basics

—part II

In part I of this series, I introduced you to the principal IC logic families and the various different forms of logic gates. Here in part II, we will continue our study of basic digital electronics by investigating *flip-flops.*

Flip-Flops

All of the digital circuits discussed thus far have operated in a "transient" manner. Gates and inverters do not have any *memory*, so once the input condition changes, then the output state that results from those conditions also is likely to change.

A flip-flop (FF) is a circuit that is capable of *storing a single bit* (i.e., a binary digit, either 1 or 0) of digital data; it will remember an input condition and hold the same output after the data has passed. There are various different types of flip-flop circuits, and they all operate on slightly different (even though similar) sets of rules. But one thing that they all have in common is the ability to store a single data bit.

All common forms of flip-flops can be made from various combinations of the basic AND, OR, NAND, NOT, NOR, and XOR gates. The NAND, NOR, and NOT gates are particularly often used to make flip-flops. Except for the two simplest flip-flops presented here in part II, most electronic circuits use IC flip-flops instead of actual IC gates. It is simply too costly to *make* flip-flops from IC gates when the same manufacturers do all of the interconnections for you by offering the various flip-flops pre-made in IC form.

Reset-Set (RS) Flip-Flops

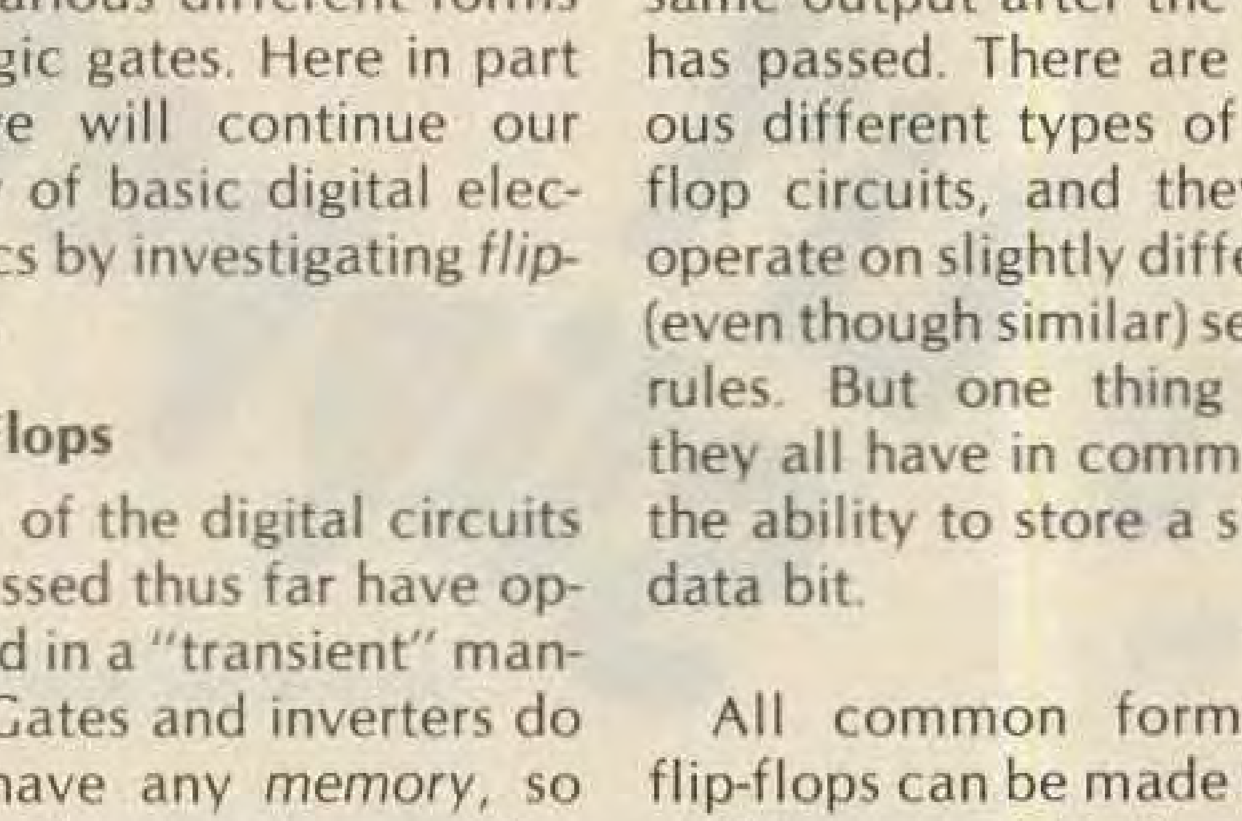
One of the simplest forms of flip-flop circuit is the *reset-set*, or *RS*, flip-flop. (Some textbooks, especially those over ten years old, call it a set-reset, or SR, flip-flop.) The RS flip-flop can be made from either two NAND gates or two NOR gates, although note that operation of the two versions is slightly different.

Fig. 1(a) shows the circuit for an RS flip-flop made from a pair of NAND gates, such as the TTL 7400 device (which contains four two-input NAND gate sections).

There are two inputs required on the RS flip-flop, *set* and *reset*. Usually there are also two output terminals, and these are complementary: Q and NOT-Q ($\overline{Q}$). *Complementary* means that one will be LOW if the other is HIGH. For example, when the Q output is HIGH, then the NOT-Q will be LOW. When the Q output is LOW, then the NOT-Q will be HIGH.

The inputs of the NAND version of the RS flip-flop are active-LOW so are sometimes designated $\overline{S}$ (NOT-S) and $\overline{R}$ (NOT-R). Whenever you see an *input* that is designated as a NOT-input, has a bar over its symbol, or that has a circle in the schematic diagram, then we know that it is an active-LOW input terminal. The circuit action of an active-LOW input occurs when the terminal is brought LOW. An example of a schematic that uses the circled inputs is shown in Fig. 1(c), while the normal symbol for the RS flip-flop is shown in Fig. 1(b).

A momentary LOW on the *set* input of the NAND gate RS flip-flop causes the outputs to go to the state where the Q is HIGH and the NOT-Q is LOW. Note that the term *set* usually means Q = HIGH and NOT-Q = LOW, while *reset* indicates just the opposite: Q = LOW and NOT-Q = HIGH. The flip-flop is said to possess *memory* (and, indeed, solid-state computer memory uses arrays of FFs), so the outputs will stay in the set condition unless a *reset* pulse is applied to the *R* input.



S	R	Q	$\overline{Q}$
0	0	DISALLOWED	
0	1	1	0
1	0	0	1
1	1	NO CHANGE	

Fig. 1. (a) A reset-set flip-flop (RS FF) can be made from two NAND gates. (b) Symbol for RS flip-flop. (c) The circled inputs for R and S indicate that these inputs are active-LOW. (d) The operation of an RS flip-flop is summarized in this truth table.

The reset function is obtained by momentarily bringing the reset input LOW. This forces the outputs to go to a state in which the Q is LOW and the NOT-Q is HIGH.

The rules for the operation of the NAND-logic RS flip-flop are summarized in the truth table shown in Fig. 1(d). This truth table also lists two additional conditions besides those discussed above. One of these is the condition in which both set and reset inputs are brought LOW simultaneously. This is a *disallowed* state, and the circuit will not know what to do; the output state will be unpredictable.

The other condition is the case where both inputs are simultaneously HIGH. In this condition we find that there is no change in the output state. The RS flip-flop simply remains in the condition present when the inputs were made HIGH.

A NOR-logic version of the RS flip-flop is shown in Fig. 2. This circuit may be constructed from TTL/7402 NOR gates. Like the 7400 device, the 7402 contains four independent two-input gates (in this case, the NOR variety). The circuit in Fig. 2 performs differently from the NAND-logic version of Fig. 1, but there are similarities even though a slightly different set of operating rules prevails.

The rules governing the NOR-logic RS flip-flop are summarized in the truth table of Fig. 2(c), but let's go over them briefly:

1) If *both* inputs are LOW, then there is no change in the output state.

2) If *both* inputs are simultaneously HIGH, then we have a disallowed state and the output condition is unpredictable.

3) If the *set* input is made HIGH momentarily, then the output condition is Q=HIGH and NOT-Q=LOW.

4) If the *reset* input is made HIGH momentarily, then the output condition is Q=LOW and NOT-Q=HIGH.

Note again the principal difference between the two forms of RS flip-flop (examine the truth tables in Figs. 1 and 2 again). The NAND-logic RS flip-flop has *active-LOW* inputs, while the NOR-logic RS flip-flop has *active-HIGH* inputs.

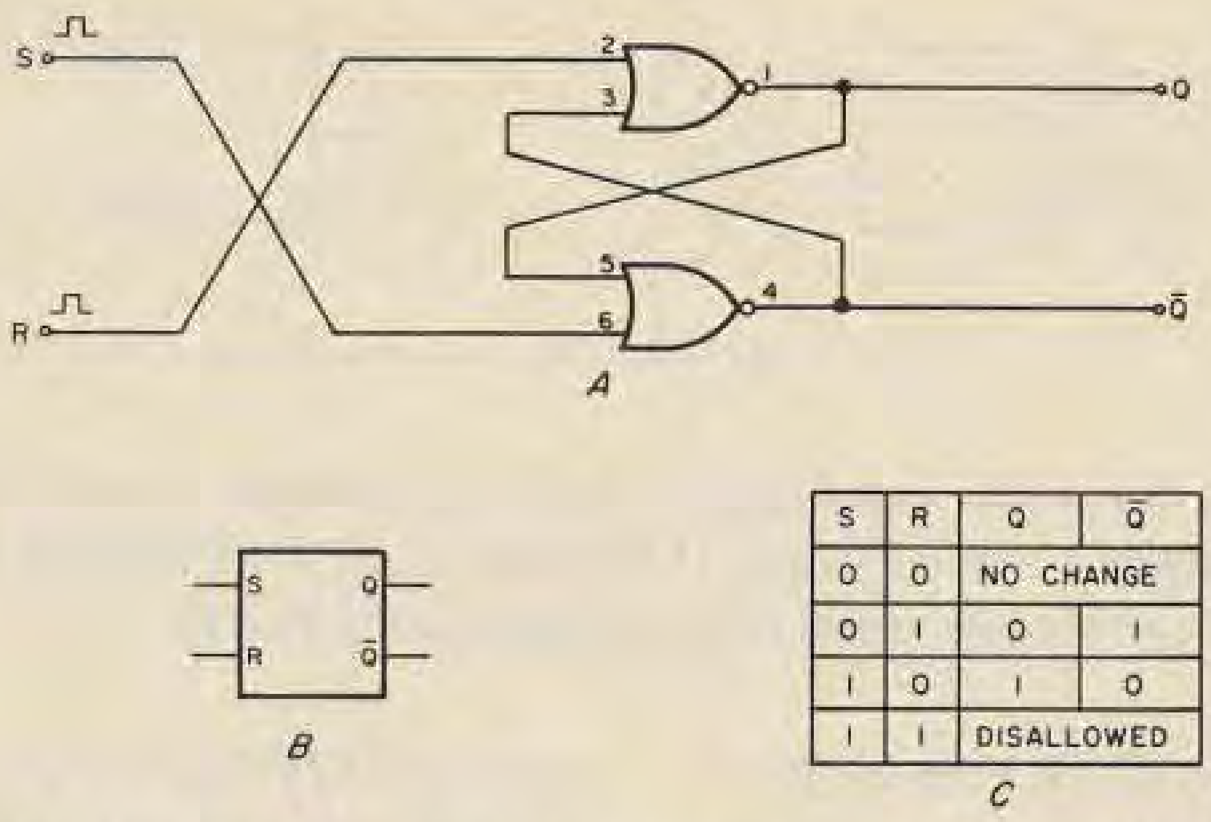
Clocked RS Flip-Flops

We sometimes get into trouble with flip-flops that are too simple. We see, for example, electronic versions of the old *relay-race* problem. In that problem and its modern electronic version with digital circuits, two relays may have slightly different actuation times. If the time difference is such that they operate out of the intended order, then catastrophic results sometimes occur. Many of these problems are solved in the digital electronics world by using *clocked,* or *synchronous,* operation. In the case of the RS flip-flop, we obtain clocked operation by using the *master-slave flip-flop,* also called the *clocked RS flip-flop.*

The purpose of the *clock* (a train of pulses) is to synchronize the changes in the output condition by allowing them to occur only at certain times during, or immediately following, a clock pulse. Most large-scale digital circuits will use synchronous operation in order to keep things straight.

There are two basic forms of clocking used in RS flip-flops: *level-triggered* and *edge-triggered.*

A level-triggered flip-flop is one in which the output state changes in response to conditions on the inputs only when the clock input is either HIGH or LOW (depending upon the type). Some level-triggered circuits require the clock pulse to be LOW for it to be active, while others (the more usual case) require the clock pulse to be HIGH.



S	R	Q	Q̄
0	0	NO CHANGE	
0	1	0	1
1	0	1	0
1	1	DISALLOWED	

Fig. 2. (a) An RS flip-flop can be made from NOR gates as well as NAND gates. (b) The RS flip-flop built from NOR gates has active-HIGH S and R inputs. (c) A NOR-logic RS flip-flop follows this truth table.

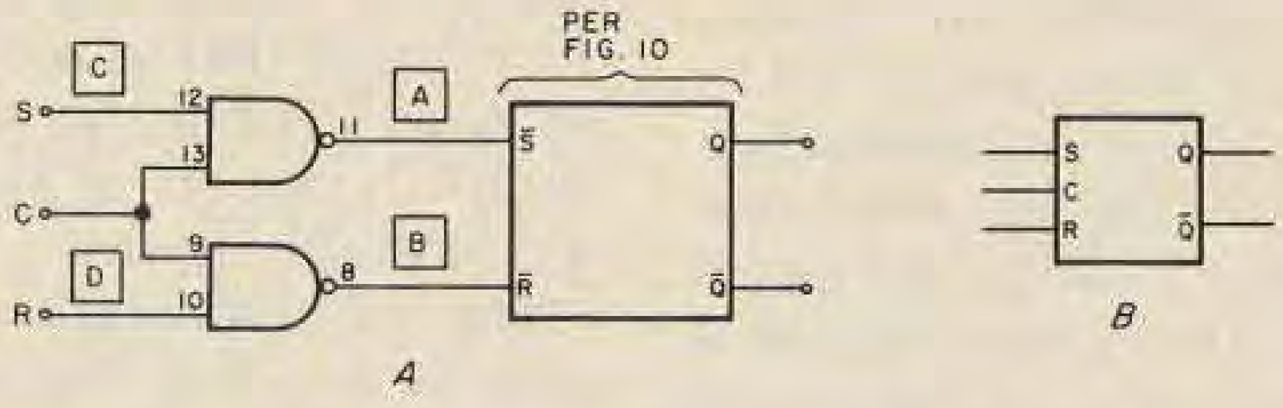


Fig. 3. (a) By adding two NAND gates to a NAND-logic RS flip-flop, a level-triggered clocked RS flip-flop is obtained. (b) Schematic symbol for a level-triggered clocked RS flip-flop.

An edge-triggered flip-flop will allow state changes only during one of the two transitions of the clock pulse. The pulse must be in the process of going from LOW to HIGH, or from HIGH to LOW (again, depending upon type). A positive edge-triggered flip-flop, therefore, will allow output changes to occur only on the positive-going transition (LOW to HIGH) of the clock pulse. A negative edge-triggered flip-flop allows output transitions only on the negative-going (HIGH to LOW) transition of the clock pulse.

It is important to remember the difference between these two types of triggering, so let's reiterate: *Level triggering* means that changes can take place only during the time when the clock pulse is active, i.e., either HIGH (positive level-triggered) or LOW (negative level-triggered); *edge triggering* means that output changes can take place only during the transition period of the clock pulse. A positive edge-triggered FF changes only on the LOW to HIGH transition, while a negative edge-triggered FF wants to see the negative-going, or HIGH to LOW, transition.

An example of a simple level-triggered clocked RS flip-flop is shown in Fig. 3. The main flip-flop is the same as the circuit in Fig. 1, so it is shown here in block form for the sake of simplicity. The *S* and *R* inputs are controlled by a pair of NAND gates. When the clock pulse is LOW, then both inputs of the RS flip-flop section (i.e., points A and B) see a HIGH, so no change can take place.

But, when the clock input goes HIGH, the levels at points A and B (i.e., the *S* and *R* inputs of the FF section) are then controlled by the other inputs of the NAND gates. These inputs are used as the *S* and *R* inputs of the clocked FF. If you doubt this, then review the operation of the NAND gates.

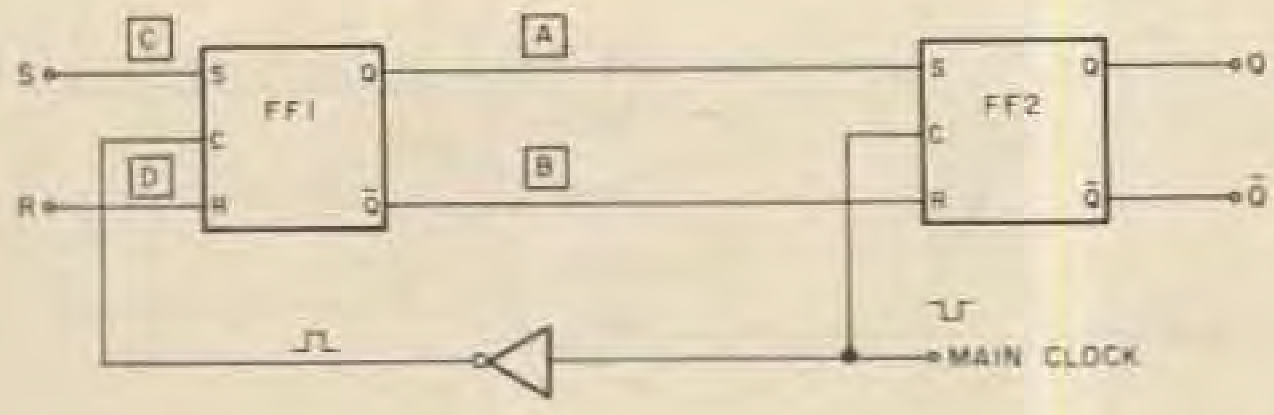


Fig. 4. Two RS flip-flops in a back-to-front configuration constitute a master-slave (M-S) flip-flop. This circuit allows only one output state change per clock pulse.

Master-Slave Flip-Flops

The use of clocking helps a great deal in taming the RS flip-flop, but several problems, again electronic versions of the old relay-race problem, still occur. Most of these are solved by using a slightly different approach—the so-called *master-slave flip-flop*. An example of the master-slave FF is shown in Fig. 4. This circuit allows only one output state change per clock pulse (the clocked RS FF allows continuous output state changes as long as the clock input is active).

The M-S flip-flop of Fig. 4 uses the clocked RS flip-flops of the previous example connected in cascade. The inverter shown allows us to drive the clock inputs of the two clocked RS FFs out of phase with each other.

Recall that the clocked RS flip-flop can change its output state only when the clock input is HIGH, and then only in response to conditions on the R and S inputs. In the M-S FF, the main clock is kept HIGH, so FF2 is active and FF1 is inactive.

When a clock pulse is applied (in this case a negative transition), FF1 will become active, and FF2 becomes inactive. Note that the effect of the inverter is to make the clock input of FF1 HIGH at this time. Any commands placed on the *S* and *R* inputs will cause changes in the outputs of FF1 (i.e., points A and B in Fig. 4).

But, because FF2 is inactive at this time (its clock input is LOW), changes at A and B are not yet reflected at the Q and NOT-Q outputs of FF2. But, once the master clock goes HIGH again, the clock input of FF2 goes HIGH again, so the changes that took place on A and B can be transferred to action at the Q and NOT-Q outputs.

The synchronization occurs by keeping FF2 inactive when the input stage (FF1) is being set up, and then rendering FF1 inactive (forbidding further *S* and *R* input changes from affecting the output), while transferring the data to FF2. This part of the sequence is called a *load-transfer operation*.

Additional Types of Flip-Flops

Thus far we have considered two versions of the RS flip-flop (NAND logic and NOR logic) and two flip-flops that are derivatives of the RS circuits, the clocked RS flip-flop and the master-slave flip-flop. In the sections to follow, we will consider some more complex types of flip-flop: *type-T* FF, *J-K* FF, and the *type-D* FF.

Type-T Flip-Flops

The type-T flip-flop (also called the *toggle FF*) is shown in Fig. 5. This FF circuit can be constructed by providing feedback connections (as shown) around an ordinary master-slave flip-flop. Recall that the M-S FF was constructed from a pair of RS FFs and an inverter stage. Note that the Q output is fed back to the reset input and the NOT-Q output is fed back to the set input.

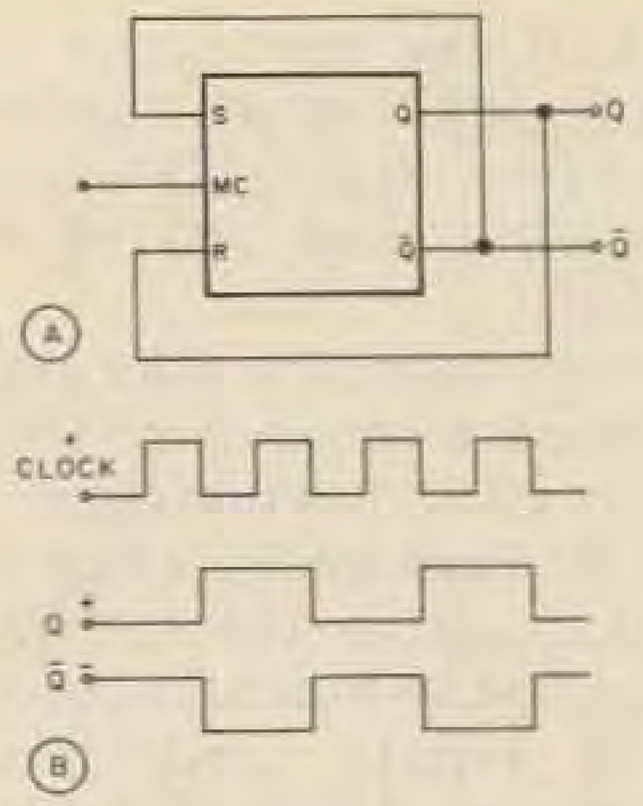


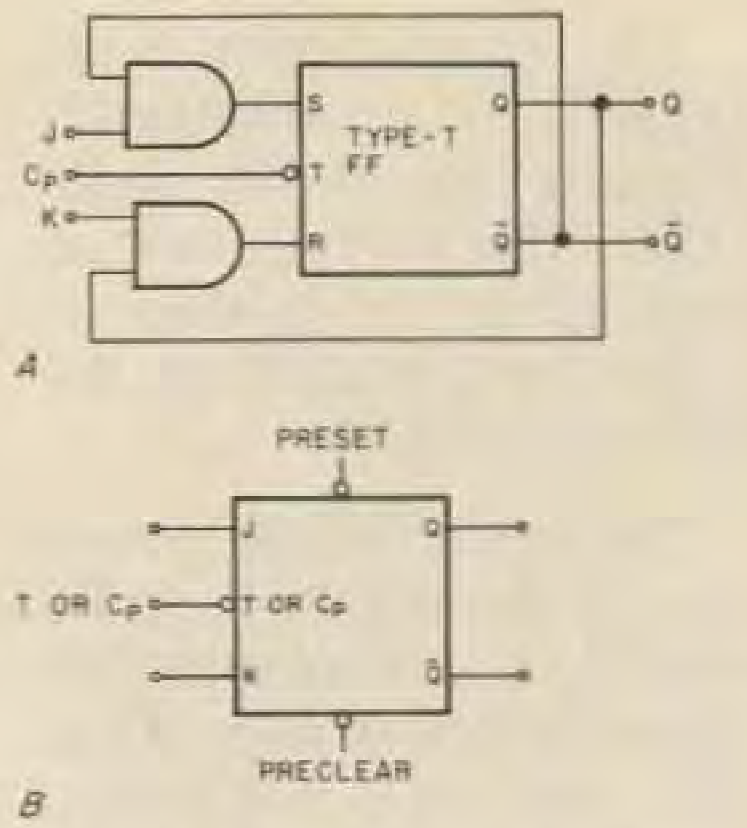
Fig. 5. (a) A type-T, or toggle, flip-flop is obtained by adding feedback connections to a master-slave flip-flop. This circuit acts as a binary divider. (b) For a toggle flip-flop, a negative-going transition of the clock results in a change of the output (Q) status.

The type-T flip-flop functions as a binary divider; that is, the output signal has a frequency that is one half (i.e., divided by 2) of the input signal. The timing diagram for this circuit is shown in Fig. 5(b). Note that the Q output changes state only on negative-going transitions of the clock pulse. At the first negative transition, the Q output will snap HIGH and remain HIGH until the clock input sees *another* negative transition. This condition occurs at pulse number 2, at which time the Q output goes LOW again. We have, therefore, binary division of the input frequency: One output pulse is produced for each *two* input pulses.

There sometimes are found differences in terminal designations from one text or spec sheet to another. In Fig. 5(a), for example, we have labeled the clock input *MC* for main clock. But it is likely that you also will see T for toggle, or C_p for clock.

J-K Flip-Flops

One of the most useful and perhaps most common forms of clocked FFs is the J-K flip-flop. There are several advantages to the typical J-K flip-flop. (a) There are no invalid or disallowed states in the clocked mode. (b) It can cause the outputs to complement. And (c), it can provide non-clocked operation (in some IC versions).



J	K	CLOCK	Q	$\bar{Q}$
0	0		NO CHANGES	
0	1		0	1
1	0		1	0
1	1		BINARY DIVISION	

C

PRESET	PRECLEAR	CLOCK	Q	$\bar{Q}$
0	0	(DOESN'T CARE)	DISALLOWED	
0	1		1	0
1	0		0	1
1	1		CLOCKED OPERATION	

D

Fig. 6. (a) Two AND gates and a type-T flip-flop combine to form a J-K flip-flop. (b) Several designations (MC, T, or Cp) are used to indicate the clock input for a J-K flip-flop. (c) When both the preset and preclear inputs are HIGH, a J-K flip-flop is in the clocked mode. The output depends on the status of the J and K inputs. (d) Direct control of the J-K flip-flop is accomplished by using the preset and preclear inputs.

Fig. 6 shows one of several popular ways to represent the J-K FF. In this case, we see that it is a type-T FF with feedback to the set and reset inputs controlled by a pair of two-input AND gates. One input from each gate accepts the feedback lines, while the remaining inputs of the gates are used to form the *J* and *K* inputs of the FF, respectively.

Fig. 6(b) shows the circuit symbol for a J-K flip-flop.

Not all versions of the J-K will have the direct-mode inputs (preset and clear). These inputs do, however, make it a more useful device. The *preset* input may also be called a *direct-set* input, and the *preclear* input called a *direct-clear* input.

Direct mode operation. The operation of the J-K flip-flop in the direct mode is very simple, and it is independent of conditions applied to the J and K inputs. The direct mode is controlled only by conditions on the *preset* and *preclear* input terminals, and the rules are summarized in Fig. 6(d).

The direct mode inputs are active when LOW, so the only disallowed state occurs when both are simultaneously LOW.

If the preset input is LOW and the preclear input is HIGH, then the outputs immediately go to a condition where Q is HIGH and NOT-Q is LOW.

If the preclear input is made LOW and the preset input is HIGH, then the outputs go to a state where Q is LOW and NOT-Q is HIGH.

It is a general rule, when dealing with flip-flops of any type, that set or preset operations make the Q output HIGH and the NOT-Q output LOW, while clear and reset operations work in just the opposite manner (i.e., Q LOW and NOT-Q HIGH).

If both preset and preclear inputs are made HIGH, then the flip-flop is ready for normal clocked operation.

Clocked operation. Whenever the preset and preclear inputs (where used) are simultaneously HIGH, the J-K will operate in the clocked mode. The rules for clocked operation are summarized in Fig. 6(c).

Like the type-T FF, the J-K FF (in the clocked mode) responds on the negative-going transition of the clock pulse. No output changes will occur regardless of changes at the *J* and *K* inputs, until one of these negative-going clock pulse transitions is seen. The outputs will then respond according to the J-K input conditions. The rules for clocked operation are as follows:

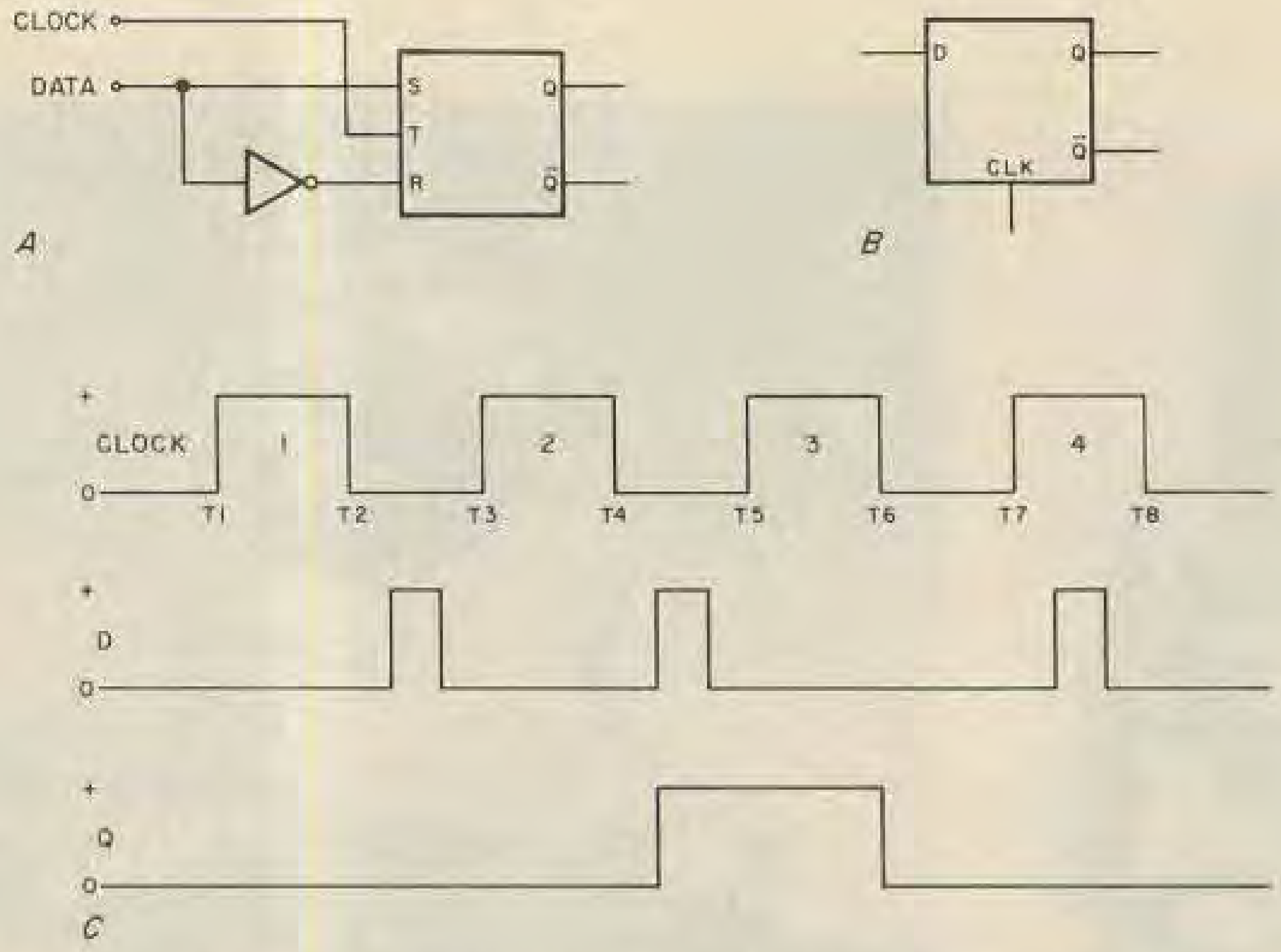


Fig. 7. (a) The type-D flip-flop is a derivation of the RS FF. (b) Symbol for a type-D flip-flop. (c) Data appearing on the D input is transferred to the Q output only when the clock line is HIGH.

1). If both J and K are LOW, then the FF is inert and does nothing. No changes will occur in the outputs.

2) If J is LOW and K is HIGH, then the clocking will make Q LOW and NOT-Q HIGH.

3) If J is HIGH and K is LOW, then the clock pulse transition makes Q HIGH and NOT-Q LOW.

4) If both J and K are HIGH, then the J-K FF behaves much like a type-T FF; clocking complements the outputs. This means that negative-going clock-pulse transitions force the outputs to go to the opposite state. The output waveform of the J-K flip-flop is then identical to the output waveform of the type-T flip-flop given in Fig. 5.

Type-D Flip-Flop

The type-D or *latch* flip-flop is shown in Fig. 7. The equivalent circuit is shown in Fig. 7(a), while the usual schematic symbol is shown in Fig. 7(b).

The equivalent circuit consists of a clocked RS FF in which the set and reset inputs are fed by the same signal but are 180 degrees out of phase with each other (i.e., complementary inputs). An inverter between the *S* and *R* lines accomplishes this neat trick.

The common line to the reset-set-inverter is called the *data* or *D* input instead of clock. This input is usually labeled D on most schematics.

The rule for operation of the type-D FF is very simple: Data appearing on the D input will be transferred to the Q output only when the *clock* line is HIGH.

1) If the clock line is HIGH, then the output will *follow* changes in the input signal (i.e., changes on the *D* input). When the D line goes HIGH, then the output will go HIGH. Similarly, when the D line goes LOW, then the outputs follow by also going LOW.

2) If the clock line is LOW, then the output will retain the last data that existed on the D input at the instant the clock line dropped LOW.

These rules can also be seen in the timing diagram of Fig. 7(c). Read the description below, keeping in mind the two rules just given.

a) When the first clock pulse arrives (T1-T2), the D input is LOW, so the Q output also will be LOW.

b) During the interval T2-T3, the D input goes HIGH, but since no clock pulse is present, it cannot affect the output conditions.

c) At the beginning of interval T3-T4, clock pulse number 2 is HIGH, but the D input is LOW. The output, therefore, must remain LOW.

d)Approximately midway through clock pulse 2, however, the D input goes HIGH, forcing the Q output to also go HIGH.

e) The Q output stays HIGH even after clock pulse 2 goes LOW.

f) At the onset of clock pulse number 3, the D input is LOW, so the Q output drops LOW also.

g) The pulse on the D input during the interval T6-T7 cannot affect the Q output because the clock is LOW.

The so-called *data latch* device is a special case of the type-D flip-flop. This device is used in digital-readout circuits (e.g., in frequency counters) to hold current data until the new data has been updated and is ready for display. This gives the illusion that the data is updated instantaneously. In most cases, the clock input is called a *strobe* input. Data at the D input will be transferred to the Q output only when the strobe line is HIGH. The idea is to bring the strobe line momentarily HIGH when the data at the input is valid, and then let the strobe line go LOW again until the next newest data is ready.

And Now . . .

The third and final part of this series, to be published next, will allow you to wade into digital electronics up past your knees. The topics will be the most common multivibrator and counter circuits. ■