DIGITAL FUNDAMENTALS

Peer into those mysterious flip-flops and see what they're made of, how they work, and how they can help simplify your designs. We'll look at the most-basic flip-flop circuits to more advanced forms of this circuit building block.

LESSON 4: Understanding and Using Flip-Flops

By L.E. Frenzel

THERE ARE TWO BASIC TYPES OF DIGITAL LOGIC CIRCUITS, *combinational* and *sequential*. Combined circuits are made up of logic gates connected in a variety of configurations. Combinational circuits typically have multiple inputs and multiple outputs. Their outputs are a function of the input states, the types of gates used, and how they are interconnected.

Sequential logic circuits also contain gates, but their main element is a logic circuit we have not yet discussed; it's called the *flip-flop*. A flip-flop is a circuit used for storing one bit of data. Because flip-flops are a kind of memory circuit, they permit a variety of storage and timing operations to be performed. Some of those operations include counting, shifting, sequencing, and delay generation.

In this lesson, you are going to learn about the various types of flip-flops and how they are used. In a future lesson, we will cover more advanced sequential logic circuits, including counters and shift registers.

Note: In the following discussion, we use the expression "high" to refer to a binary-1 logic level of some positive voltage in the +3 to +5 volt range. "Low" is used to designate a binary-0 logic level, which is ground or 0 to +.2 volt.

Data Latches

The simplest form of flip-flop is the latch or RS flip-flop. Like all other flip-flops, this type is capable of storing one bit of data. It has two inputs and two outputs, and is usually represented by the simple logic block shown in Fig. 1. For example: To store a binary 1, you apply a signal momentarily to the set input. To store a binary 0 in the latch, you momentarily apply a logic signal to the reset input. Once the latch is set or reset by the input pulse, it remains in that state. The flip-flop remembers to which state it was set (0 or 1) until the state is changed, or until power to the circuit is removed.

To determine what bit is stored in the latch, you look at the outputs. By examining the normal output with a voltmeter, logic probe, or oscilloscope, you can determine what state the flip-flop is in. If the normal output is a binary 1, then the flip-flop is set and storing a binary 1. If the normal output is binary 0, the flip-flop is reset and a binary 0 is being stored. The complementary output is an inverted version of the normal output and is useful when the latch is used to drive other logic circuits.

Incidentally, you will note that in Fig. 1, the outputs of a

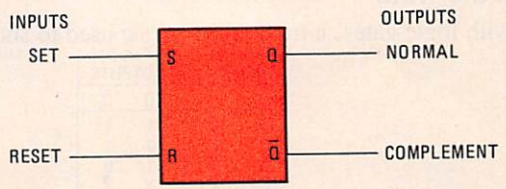

Fig. 1—The logic symbol for a RS (reset-set) flip-flop or latch is given above.

flip-flop are normally labelled with letters of the alphabet. Q is commonly used with flip flops, but other letters of the alphabet or other mutli-letter combinations can be used, too. Also, when a line or a bar appears over a letter as in $\overline{Q}$, that signal is the reverse of its counterpart, Q. That is to say, should Q be at a low, then $\overline{Q}$ will be at a high.

A latch or RS flip-flop is easily constructed with NAND gates as shown in Fig. 2. We say that the gates are connected back to back with the output of one connected to the input of the other, and so on. The operation of a latch is easy to understand if you remember how a NAND gate works. The simple truth table in Fig. 2 will refresh your memory.

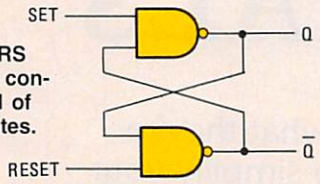

Fig. 2—RS flip-flop constructed of NAND gates.

TRUTH TABLE OF NAND GATE

INPUTS		OUT
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

When power is first applied to a flip-flop circuit, the latch comes up in one of its two stable states. Because of minor differences between the two gates, the circuit will flip to either the set or reset state immediately upon power-up. You will not be able to predict which state will occur. Let's assume that the flip-flop initially comes up in its set state. That means that the Q output is a binary 1. The binary 1 appears at the input of gate 2 along with the reset input. The reset input is shown open here. Remember, that an open input usually has the same effect as a binary-1 input. With those conditions on gate 2, its output will be a binary 0. The output of gate 2 is applied back to the input of gate 1. The set input is also open at that time and has the effect of a binary 1. However, it has no effect on the circuit, because the binary-0 input to gate 1 causes its output to remain high.

Just to be sure you understand the idea, trace the circuit state by assuming that the flip-flop comes up in the reset condition. Start out with the complement output being binary 1 and repeat the analysis above.

Keep in mind that the set and reset inputs will normally not be open. Instead, they will be held at the binary-1 level. To change the state of the flip-flop, either the set or reset input must be brought momentarily to the binary-0 level.

Assume that the flip-flop is initially set with the Q output being binary 1. If we want to reset the latch, we simply apply a brief pulse that switches from binary 1 to binary 0 and back again. The binary-0 input on gate 2 immediately forces its output high. That high output to the input of gate 1 along with the high set input causes the normal output to go low. The flip-flop changes state from set to reset.

Incidentally if another reset pulse is applied to the reset input, no additional state change will occur. If the flip-flop is already reset, then another reset input pulse will have no effect on the circuit. The same is true for set pulses.

To Tell the Truth

As with logic gates, a truth table can be used to show all possible states of a latch. The truth table in Fig. 3 shows the various combinations of inputs and outputs. There are two special conditions that we should explain. First, when both inputs are binary 1, the state of the latch is not affected. Since we don't know which state the latch is in, we simply designate the output with the letter X—which, of course, can represent 1 or 0.

INPUTS		OUTPUTS	
S	R	Q	$\overline{Q}$
0	0	1	1*
0	1	0	1
1	0	1	0
1	1	X	X

X = EITHER 0 OR 1
* = AMBIGUOUS STATE

Fig. 3—Truth table for a RS flip-flop.

Another special condition occurs when both inputs are binary 0. At that time both outputs will be forced to the binary-1 level. Looking at the normal output, you will see a binary 1 and, therefore, might suspect that the flip-flop is set. However, that is not the case, because as the complement output is also a binary 1, implying that reset is an ambiguous state that does not represent either the set or reset condition. It should be avoided by eliminating the possibility that both inputs could go to binary 0 simultaneously.

The operation of the latch can also be illustrated with input and output waveforms as shown in Fig. 4. Take a minute to look over those signals to be sure you understand the operation of a flip-flop. The way to do it is simply to observe the Q or normal output to determine the state of the flip-flop. Then note how the set and reset inputs change it. The complement output, of course, is simply an inversion of the normal output except in the ambiguous state.

You can also construct a latch using NOR rather than NAND gates. A NOR latch is shown in Fig. 5. The flip-flop has normal and complement outputs, but note that the positions of the set and reset inputs have been reversed. Because the operation of a NOR is different from that of a NAND, the signals used to change the state of the flip-flop must be binary 1 rather than binary 0, as with NAND gates. Normally both the set and reset inputs will be binary 0. At that time, the flip-flop will be either in its set or reset state. To change the state of the flip-flop, a momentary binary-1 pulse is applied to either the set or reset input. Figure 6 shows the truth table for a NOR latch. The operation of that circuit is further described by the timing waveforms shown in Fig. 7.

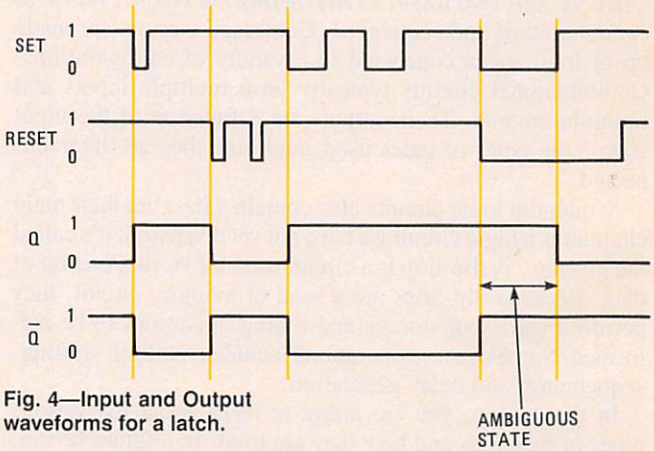

Fig. 4—Input and Output waveforms for a latch.

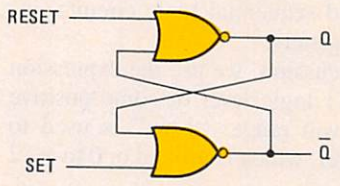

Fig. 5—RS flip-flop constructed of NOR gates.

INPUTS		OUTPUTS	
SET	RESET	Q	$\overline{Q}$
0	0	X	$\overline{X}$
0	1	0	1
1	0	1	0
1	1	0	0*

X = EITHER 0 OR 1
* = AMBIGUOUS STATE

Fig. 6—Truth table for the NOR latch.

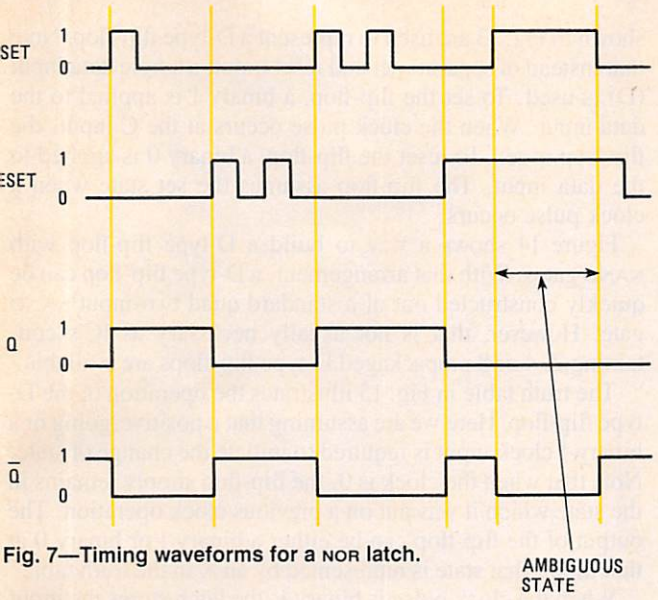

Fig. 7—Timing waveforms for a NOR latch.

Debounce

A popular application for a latch is switch *debouncing*. Switch debouncing is the process of removing the noise (undesired electrical signal) created by most switches when they are opened or closed. Switches of all types are widely used to create binary input signals for logic circuits. A typical arrangement is shown in Fig. 8. With the switch open, the output is a binary-1 level as seen through the resistor. When the switch is closed, the output is brought to ground or binary 0. While clean, clearly defined logic levels are generated by that simple circuit, the problem lies in the garbage generated by the switch for a brief duration when it is opened or closed. The waveforms in Fig. 8 illustrate that effect. Whenever two metal contacts are opened or closed, they will often vibrate or not cleanly make or break contact for a short duration. Any dirt or other foreign matter on the contacts will aggravate the problem. The result is multiple pulses or spikes during opening or closing. Such noise can falsely trigger logic circuits. A latch can be used to eliminate the problem.

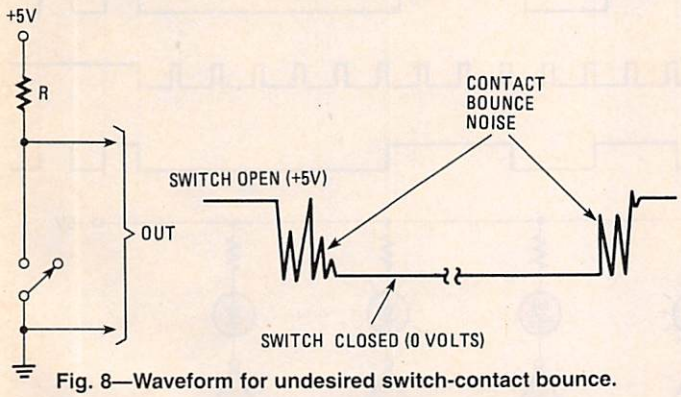

Fig. 8—Waveform for undesired switch-contact bounce.

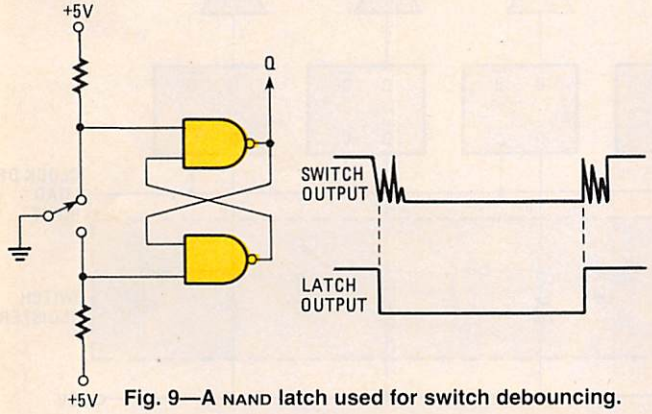

Fig. 9—A NAND latch used for switch debouncing.

Fig. 9 shows a latch debounce circuit. A single-pole, double-throw (SPDT) switch must be used in the application. While the contacts still bounce at the inputs to the latch, they effectively have no affect on the output. Recall that if a signal is repeatedly applied to the set or reset input, the flip-flop state will not change. Once it is set, it remains set despite additional inputs. Once it is reset, it remains reset despite erroneous inputs. The result is an output signal that follows the switch conditions, but whose transitions from 0 to 1 and 1 to 0 are *clean*.

A NOR latch can also be used for switch debouncing, as shown in Fig. 10. However, note that the switch input is +5 volts, or a binary-1 level, rather than ground as in the NAND latch. Otherwise the operation of the circuit is similar.

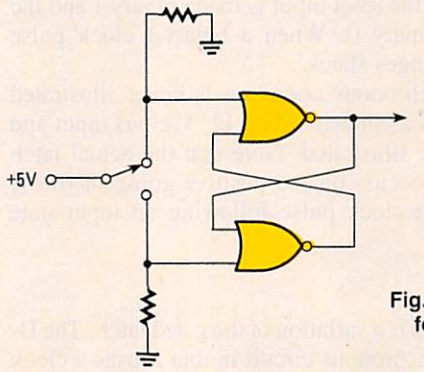

Fig. 10—A NOR latch used for switch debouncing.

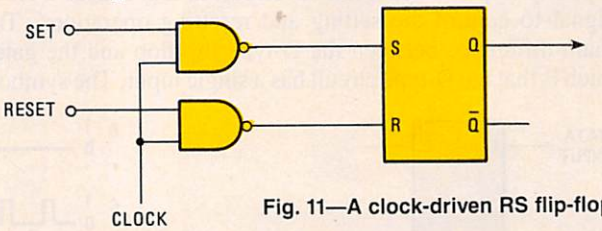

Fig. 11—A clock-driven RS flip-flop.

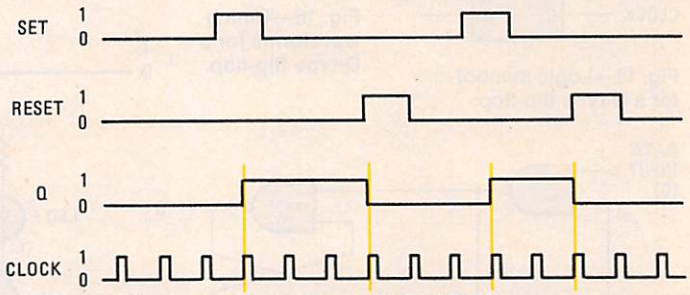

Fig. 12—Timing waveforms for a clocked latch.

Clocked RS Flip-flop

The latch or RS flip-flop is an *asynchronous sequential circuit*. That means that the output changes state immediately upon application of the input signals. On the other hand, some logic circuits act in response to a master timing signal called a *clock*. A clock is an oscillator circuit that generates a fixed-frequency periodic sequence of pulses that are used to control all timing and sequencing operations in a digital cirucit. Logic circuits controlled by a clock are referred to as synchronous because all state changes are intiated and occur in step with the clock signals. Asynchronous circuits, of course, do not use a clock and state changes occur immediately upon applications of inputs. Clocked logic circuits are

more predictable and are generally immune to "race" conditions that exist in some asynchronous circuits.

A basic latch can be modified to be used in synchronous operations as shown in Fig. 11. Here the set and reset inputs are buffered by NAND gates. The operation of those NAND gates is controlled by the clock. It is assumed that the latch is a NAND latch whose set and reset inputs must momentarily be switched to the binary-0 condition to cause a state change.

To set the latch, a binary 1 is applied to the set input and a binary 0 is applied to the reset input. With those inputs, the latch does not change state immediately. The reason for that is that the clock is normally in the low position. That inhibits the NAND gates keeping their outputs high and the latch unaffected. When a binary-1 clock pulse occurs, the NAND gates are enabled and the set and reset input signals are applied to the S and R inputs of the latch. The S input goes low while the R input remains high. The result is that the latch is set and the Q output goes to binary 1.

To reset the latch, the reset input is made binary 1 and the set input is made binary 0. When a binary-1 clock pulse occurs, the latch changes states.

That form of synchronous operation is better illustrated with timing diagrams as shown in Fig. 12. Various input and output conditions are illustrated. Note that the actual latch output state change occurs on the positive going or 0 to 1 transition of the first clock pulse following an input-state change.

D-Type Flip Flop

The D-type flip-flop is a variation of the gated latch. The D-type flip-flop is a synchronous circuit in that it uses a clock signal to control the setting and resetting operations. The main difference between the D-type flip-flop and the gated latch is that the D-type circuit has a single input. The symbols shown in Fig. 13 are used to represent a D-type flip-flop. Note that instead of separate set and reset input, a single data input (D) is used. To set the flip-flop, a binary 1 is applied to the data input. When the clock pulse occurs at the C input, the flip-flop is set. To reset the flip-flop, a binary 0 is applied to the data input. The flip-flop assumes the set state when a clock pulse occurs.

Figure 14 shows a way to build a D-type flip-flop with NAND gates. With that arrangement, a D-type flip-flop can be quickly constructed out of a standard quad two-input NAND gate. However, that is not usually necessary as IC's containing 2, 4 or 8 prepackaged D-type flip-flops are available.

The truth table in Fig. 15 illustrates the operation of the D-type flip-flop. Here we are assuming that a positive-going or a binary-1 clock input is required to initiate the change of state. Note that when the clock is 0, the flip-flop simply remains in the state which it was put on a previous clock operation. The output of the flip-flop can be either a binary 1 or binary 0 at that time. That state is represented by an X in the truth table.

When the clock pulse is binary 1, the latch stores the input state. If the input is binary 1 while the clock is high, the latch will set and its output will be binary 1. If the input is binary 0 while the clock is high, the latch will be reset and the normal output will be binary 0. Keep in mind that while the clock input is high, the normal output directly follows the signal applied to the D input. Ordinarily the clock only occurs for a very short interval. Because of the input gating circuits, the ambiguous state cannot occur in a D-type flip-flop.

The waveforms in Fig. 16 summarize the operation of the D-type flip-flop. All possible combinations of inputs and outputs shown in the truth table are repeated in the timing diagrams. Take a look through them to confirm your knowledge of the circuit's operation.

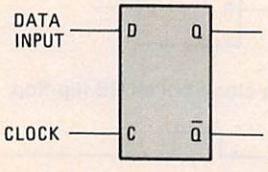

Fig. 13—Logic symbol for a D-type flip-flop.

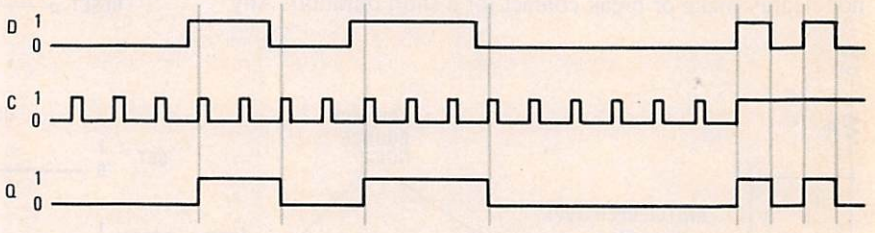

Fig. 16—Timing waveforms for a D-type flip-flop.

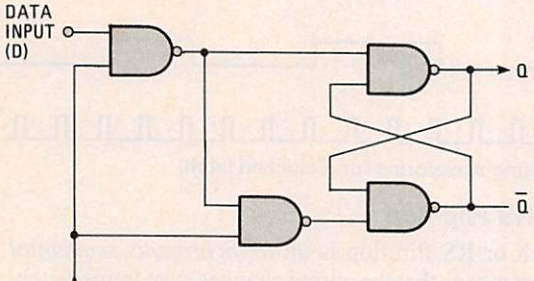

Fig. 14—A D-type flip-flop made from a quad 2-input NAND integrated-circuit chip.

D	C	Q
0	0	X
0	1	0
1	0	X
1	1	1

Fig. 15—Truth table for a D-type flip-flop.

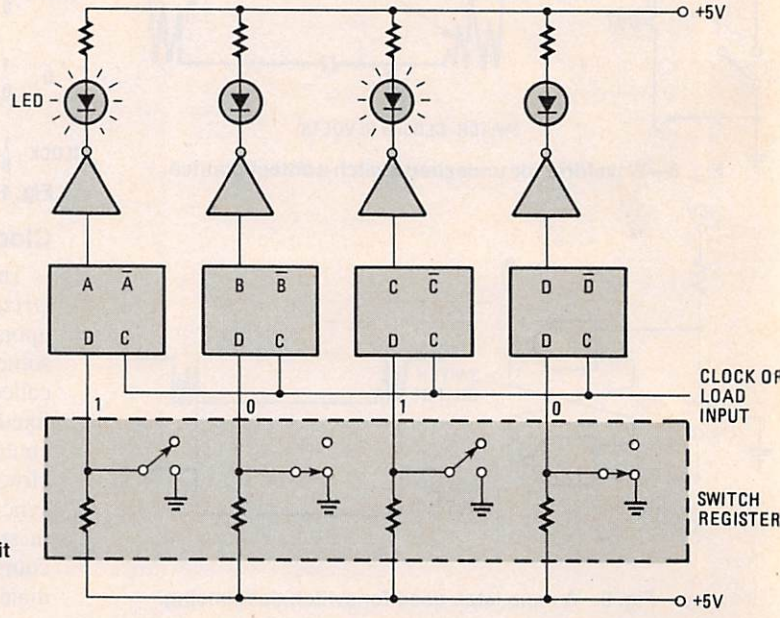

Fig. 17—A four-bit storage register.

Storage Registers

One of the main uses for D-type flip-flops is to form *storage registers*. A storage register is a circuit capable of storing a binary word. One flip-flop is needed for each bit in the word. For example, to store one byte of data, eight types of flip-flops would be needed.

Fig. 17 shows a storage register for a four-bit word. The parallel inputs to the register are supplied by a set of switches referred to as a switch register. The switch register allows you to manually select a binary word to be stored in the register. The output of the register drives light-emitting diode (LED) driver circuits. The LED's indicate the binary flip-flop states.

To store or load a word into the register, the desired word is set manually with the switches. Then a clock pulse is applied to the load input. The word will be stored in the D flip-flops and their state will be shown by the LED indicators. Note that the A flip-flop is designated as the most significant bit (MSB), while flip-flop D is the least significant bit (LSB). Therefore, the word stored is 1010.

JK Flip-flops

The most sophisticated and versatile form of storage circuit is the JK flip-flop. It can perform the functions of both RS and D type flip-flops. But it also has some additional unique features of its own. The JK flip-flop is widely used to form storage registers, but finds its greatest application in implementing sequential logic circuits such as counters and registers. You will learn more about those circuits in a future lesson.

The symbol used to represent a JK flip-flop is shown in Fig. 18. We won't discuss the internal logic circuits of a JK flip-flop, because they are somewhat complex. Besides, you don't really need to know what's inside to understand its operation or to use it.

The JK flip-flop has five inputs and two outputs. The S and C inputs, meaning "set" and "clear," are similar in operation to the set and reset inputs on a basic latch. Clear is the same as reset as it puts the flip-flop in the binary 0 or reset state.

The J and K inputs are synchronous inputs similar to the set and reset inputs on a gated latch. "J" means set while "K" means reset. The T input is for the clock. Finally, standard normal and complement outputs are generally provided.

The S and C inputs are asynchronous in nature. Those inputs are normally held high and in that state have no effect on the operation of the flip-flop. However, to set or reset the flip-flop as you would an ordinary latch, momentary low signals are applied as needed. For example, to reset the flip-flop, a binary-0 pulse would be applied to the C input momentarily. The normal output would go to the binary-0 state.

The truth table in Fig. 19 illustrates the effect that the S and C inputs have on the outputs. The results are identical to those obtained with the NAND latch discussed earlier. It is necessary to avoid the condition where both S and C inputs are low, so that the ambiguous state will be avoided.

The asynchronous S and C inputs override the J, K, and T synchronous inputs. Their effect is immediate and they predominate over synchronous operations.

The main application for the S and C inputs is presetting. To preset a flip-flop means to put it into one state or another prior to another operation taking place. An example is the resetting of a storage register. Resetting or clearing a register means setting all of the flip-flops to the binary-0 state. That would be done by connecting all the C inputs of the flip-flops together and applying a low pulse. The register is then said to be cleared.

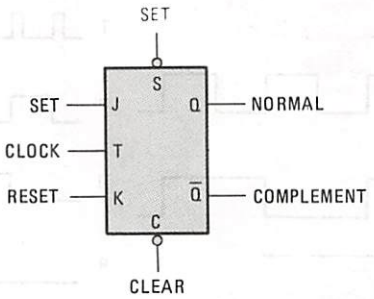

Fig. 18—Logic symbol for a JK flip-Flop.

INPUTS		OUTPUTS	
S	C	Q	Q̄
0	0	1	1*
0	1	1	0
1	0	0	1
1	1	X	X̄

X = EITHER 1 OR 0
* = AMBIGUOUS STATE

Fig. 19—Truth table for the S and C inputs of a JK flip-flop.

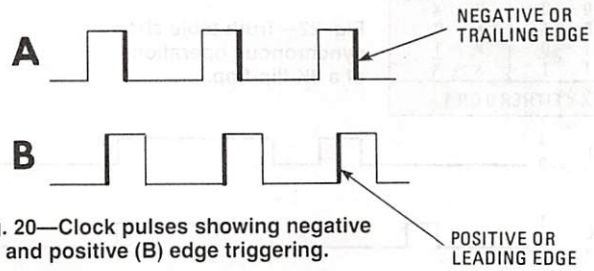

Fig. 20—Clock pulses showing negative (A) and positive (B) edge triggering.

Presetting can also mean setting the flip-flop. Occasionally it is necessary to load a specific binary number into a register prior to another operation beginning. By the use of external gates connected to the S and C inputs, any binary number can be preloaded into the register.

Now let's consider the synchronous inputs. As in a gated latch, the J and K inputs are used to set and reset the flip-flop but under the control of a clock pulse. If the J input is made binary 1 and the K input binary 0, the flip-flop will be set when the clock pulse occurs. If the J input is a binary 0 and the K input is a binary 1, the flip-flop is reset on the occurance of the clock pulse. In most JK flip-flops, that state change occurs on the trailing or negative edge of the clock signal; it is illustrated in Fig. 20-a. Some kinds of flip-flops initiate a set or reset operation on the positive or leading edge of the clock signal as shown in Fig. 20-b. Negative edge triggering, however, is more common.

When both the J and K inputs are held at binary 0, nothing happens. Even when a clock pulse occurs, no state change occurs. The flip-flop simply remains in the state in which it was previously set.

When both the J and K inputs are binary 1, an unusual action occurs. When a clock pulse appears, the flip-flop will be toggled or complemented. What that means is that on the trailing edge of the clock pulse, the flip-flop will simply change state. If the flip-flop was set on the occurence of the trailing edge of the clock pulse. That unique feature of the JK flip-flop allows it to be used in a variety of counter and frequency divider circuits as you will see. Figure 21 illustrates that toggling or complementing mode of operation.

The synchronous operation of the JK flip-flop is summarized by the truth table in Fig. 22. The four possible combinations of the JK inputs are shown. Note that the output is expressed in two ways. First, the Q_n column is the normal output state of the flip-flop. Note that all of the entries in that column are designated X which means that the flip-flop may be either set or reset. The other output column is designated $Q_n + 1$. That is also the normal output, but it designates the

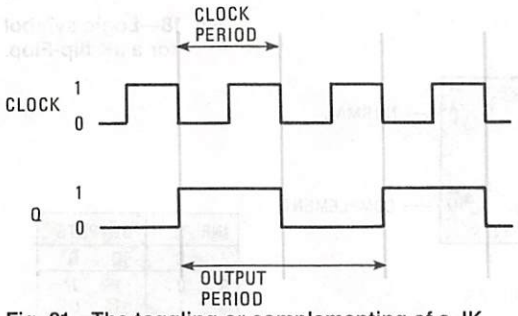

Fig. 21—The toggling or complementing of a JK flip-flop by a clock when the JK inputs equal 1.

INPUTS		OUTPUTS	
J	K	Q	Q n+1
0	0	X	X
0	1	X	0
1	0	X	1
1	1	X	$\overline{X}$

X = EITHER 0 OR 1

Fig. 22—Truth table showing synchronous operation of a JK flip-flop.

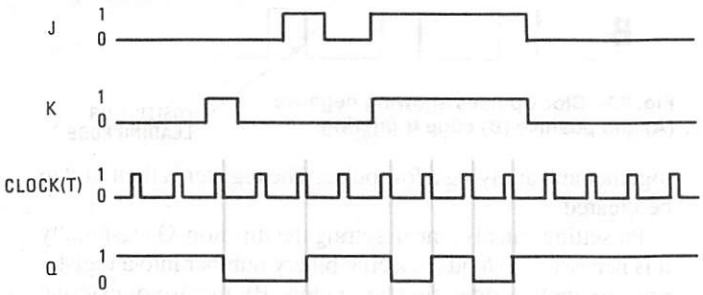

Fig. 23—Synchronous timing waveforms of a JK flip-flop.

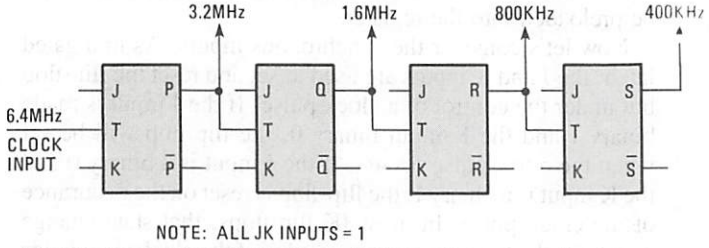

Fig. 24—Cascading JK flip-flops to form a frequency divider.

state of the flip-flop after the occurance of a clock pulse with the designated JK inputs.

Figure 23 shows the timing waveforms of a JK flip-flop. Work your way through those diagrams from left to right to be sure that you understand all conditions—trailing-edge triggering is assumed.

JK Flip-flop Applications

As indicated earlier, the JK flip-flop finds its greatest use in various kinds of registers and counters. We won't discuss those here as a complete lesson is devoted to them later. However, we do want to illustrate several simple applications.

A major application of JK flip-flops is a frequency divider. Refer to the input and output signals of a typical JK flip-flop as shown previously in Fig. 21. Note that each time a negative going transition occurs, the flip-flop will toggle. Because of that, the output of the flip-flop is one-half the frequency of the input. Or the output period is twice the period of the clock. We say that the flip-flop is a divide-by-2 circuit. If a 100-kHz clock is applied to the flip-flop, the output will be a 50-kHz signal.

JK flip-flops can be cascaded to perform frequncy division by greater multiples of 2 (4, 8, 16, 32, etc.). In Fig. 24 we show four JK flip-flops cascaded with the normal output of

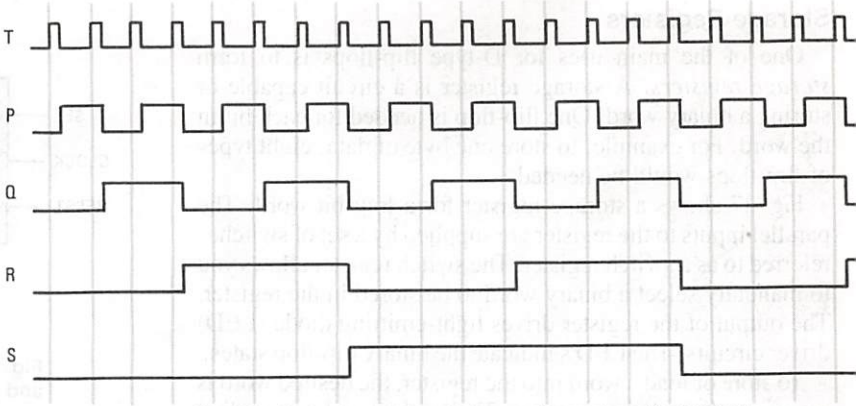

Fig. 25—Output waveforms from a four-stage frequency divider. Negative edge triggering is used.

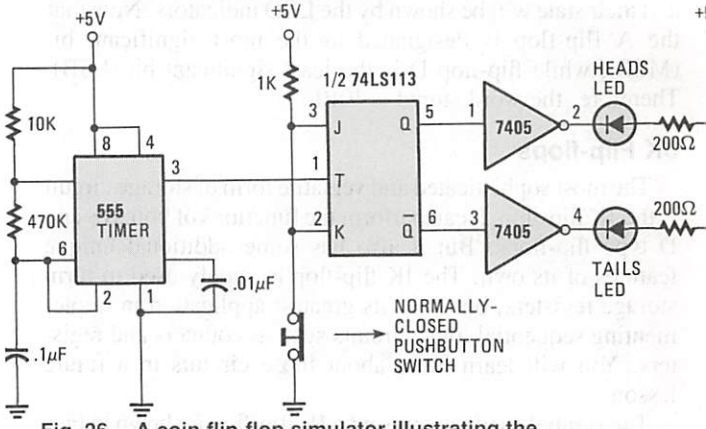

Fig. 26 —A coin flip-flop simulator illustrating the operation of a JK flip-flop.

one connected to the T input of the next. Naturally each flip-flop divides by 2. With the 6.4-MHz input shown, the other flip-flop outputs are 3.2-MHz, 1.6-MHz, 800-kHz and 400-kHz as shown. The waveforms in Fig. 25 show the full operation of the circuit.

An easy way to determine the division factor for a given number of flip-flops is to use the simple relationship shown below. Here F represents the frequency division ratio or factor. It is equal to 2 raised to the *n* power where *n* is the number of flip-flops in the chain. With four flip-flops, the frequency division ratio is:

$$F = 2^n = 2^4$$

$$F = 2 \times 2 \times 2 + \times 2$$

Learn By Building

Figure 26 shows a simple circuit you can build to understand the operation of a JK flip-flop. Here a 555 timer IC is connected as a clock. It generates a clock signal that will repeatedly toggle the JK flip-flop whenever the pushbutton switch is depressed. When the switch is released, the JK inputs are held low and the clock has no effect on the flip-flop.

The outputs of the JK flip-flop are connected to LED driver circuits. You will find that the outputs will always be complementary as indicated by one LED being on while the other is off.

That circuit simulates the flipping of a coin. For example, heads might represent set while tails indicates reset. To flip the coin, all you do is press the pushbutton switch. The JK inputs go high. The flip-flop will then toggle repeatedly for a period of time. When you release the pushbutton, the JK inputs go low. The flip-flop will then be set or reset depending

(Continued on page 94)

(Continued from page 90)

upon where the flip-flop was prior to releasing the switch. Because of the high-speed nature of the clock, and the random depressing and releasing of the pushbutton, the circuit accurately simulates the random flipping of a coin. Another way to put it: No matter how hard you try, you can't fix a head or tail setting by holding the switch closed a practiced time interval. ■

SHORT QUIZ ON DIGITAL FUNDAMENTALS—LESSON 4: UNDERSTANDING AND USING FLIP-FLOPS

1. When a flip-flop is storing a binary 1, it is said to be:

a. reset b. set

2. Another name for a latch is _____.

3. A common application of a latch is_____.

4. To clear a flip-flop means to:

a. reset it to 0 b. preset it to 1

5. Flip-flops such as the D and JK types whose state changes occur upon the occurrance of a clock pulse are said to be_____.

6. When the JK inputs are 0 and a clock pulse occurs, the flip-flop will:

a. set b. reset

c. toggle d. not change state

7. When the JK inputs are 1 and the clock pulse occurs, the flip will:

a. set b. reset

c. complement d. remain in the same state

8. The clock input to a D flip-flop is high. The D input is low. The complement output will be _____.

9. A six bit register is made up of D-type flip-flops. The flip-flops are labelled A through F with A being the LSB and F being the MSB. The flip-flop outputs are A = low, B = high, C = high, D = low, E = high, F = high, where high = 1 and low = 0. The decimal equivalent of the binary number stored in the register is _____.

10. A frequency divider made up of 7 cascaded JK flip-flops is used to generate an output frequency of _____ kHz with an input of 512 kHz.

ANSWERS TO THE ABOVE QUESTIONS

1. b-set
2. RS flip-flop
3. switch or contact debouncing
4. a-reset it to 0
5. synchronous
6. d-not change state
7. d-complement or toggle
8. high. With the clock input high, the normal output will be the same as the D input. The complement output will be inverted.
9. 54 (FEDCBA = 110110 = 54)
10. 4kHz (7 flip-flops divide by $2.2^7 = 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2$ = 128. 512 kHz ÷ 128 = 4 kHz.