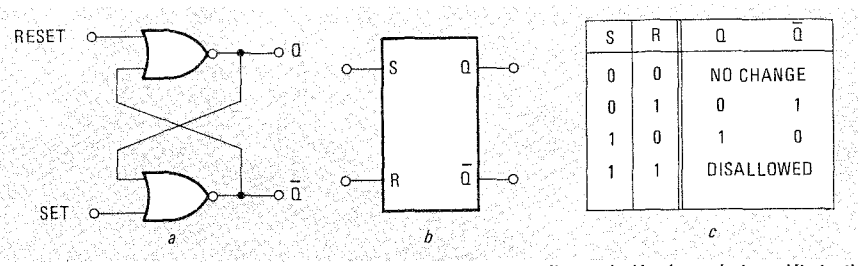
WORKING WITH FLIP-FLOPS

Flip-flops are the basis of all digital circuits. Learn about the different types and practical applications for them.

RAY MARSTON



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

FIG.1—THE SET-RESET FLIP-FLOP is built from two NOR gates *a*. Its symbol is shown in *b*, and its truth table in *c*.

DIGITAL IC'S CAN BE CLASSIFIED INTO TWO basic types: gates and flip-flops. The latter are also known as bistable latches and memory elements. Many devices are based on flip-flops, including counters, dividers, shift registers, data latches, etc., as well as presettable up/down counters and dividers, and other devices.

In this article we will explain how several types of flip-flops work. Then we'll go on to discuss several versatile CMOS flip-flops. Last, we'll show many practical circuits that use flip-flops.

Basic principles

The simplest type of CMOS flip-flop is the cross-coupled bistable latch shown in Fig. 1-*a*. The circuit is built from two NOR gates; it has two inputs (usually tied low via pull-down resistors), and a pair of out-of-phase outputs. The circuit works like this: If the SET terminal is briefly taken high, the Q output immediately goes high, and the $\overline{Q}$ output goes low. The cross-coupling between the two gates causes the outputs to latch in that state, even when both inputs are pulled low again. The only way the output states can be changed is by applying a high to the RESET terminal, in which case the Q output immediately goes low, and the $\overline{Q}$ output goes high. Again, cross-coupling causes the outputs to latch into the new state even when both inputs are pulled low.

Because of the latching action, the basic Set-Reset (S-R) flip-flop acts as a simple memory element that "remembers" which of the two inputs last went high. Note, however, that the output state cannot be predicted if both inputs go high simultaneously, so that must not be allowed to occur. Fig. 1-*b* shows the symbol of the S-R flip-flop, and Fig. 1-*c* shows its truth table.

The versatility of the basic circuit can be enhanced greatly by wiring an AND gate in series with each input terminal as shown in Fig. 2. That way high input signals can reach the S-R flip-flop only when the clock (CLK) signal is also high. Therefore, when CLK is low, both inputs of the flip-flop are held low, irrespective of the states of the SET and RESET inputs, so the flip-flop functions as a "permanent" memory. However, when CLK is high, the circuit functions as a standard S-R flip-flop. Consequently, information is not automatically latched into the flip-flop, but must be "clocked" in; that's why the circuit is known as a clocked S-R flip-flop.

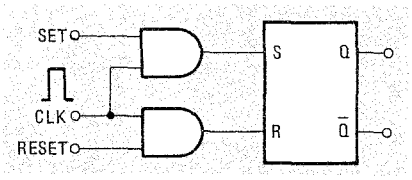


FIG. 2—THE CLOCKED S-R FLIP-FLOP is built from two AND gates, in addition to the S-R flip-flop.

Figure 3-*a* shows how to make the most important of all flip-flops, the clocked master-slave flip-flop. It's built from two clocked S-R flip-flops that are cascaded and clocked out of phase via an inverter in the clock line.

It works as follows. When the CLK input is low, the inputs to the master flip-flop are enabled via the inverter, so the SET-RESET data is accepted. However, the inputs to the slave flip-flop are disabled, so the data is not passed to the output terminals. Then, when the CLK input goes high, the inputs to the master flip-flop are disabled, so the input data is latched in the outputs; simultaneously, the input to the slave flip-flop is enabled, and the latched data is passed to the output terminals. The symbol of the clocked master-slave flip-flop is shown in Fig. 3-*b*.

The clocked master-slave flip-flop can be made to toggle (or divide by two) by cross-coupling the input and output terminals as shown in Fig. 4-*a*. By doing so, SET and Q (and RESET and $\overline{Q}$) are always at opposite logic levels. So when CLK goes

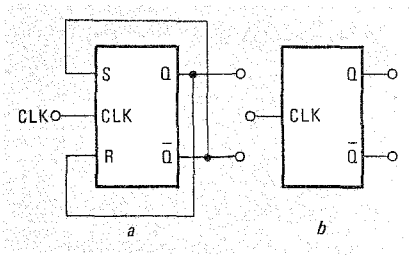


FIG. 4—THE TOGGLE OR TYPE-T FLIP-FLOP (*a*) is built from a clocked master-slave flip-flop. Its symbol is shown in *b*.

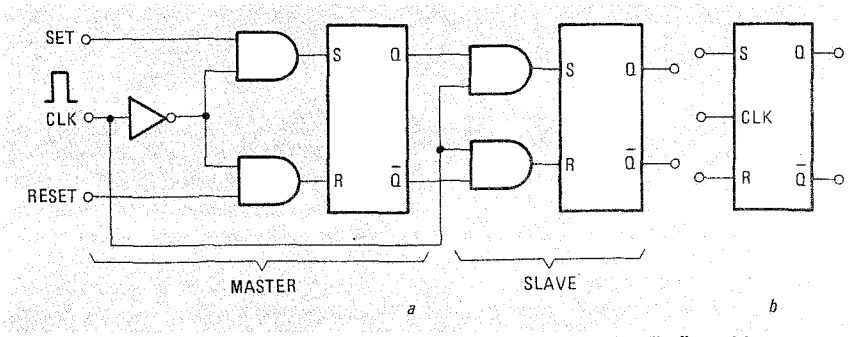


FIG. 3—THE CLOCKED MASTER-SLAVE FLIP-FLOP (*a*) is built from two S-R flip-flops driven by an out-of-phase clock. Its symbol is shown in *b*.

low, the master flip-flop changes state. When CLK goes high, the slave flip-flop changes state. Note that the output states change on the arrival of the leading edge of each new clock pulse.

It takes two clock pulses to change the output from one state to another and back again, so the frequency of the output is half the frequency of the clock. The circuit is known as a Toggle (or type-T) flip-flop; its symbol is shown in Fig. 4-*b*.

The D flip-flop

The type-T flip-flop is a special device that functions only as a counter/divider. A far more versatile device is the Data or type-D flip-flop, which is made by connecting the clocked master-slave flip-flop as shown in Fig. 5-*a*. In that circuit, an inverter is wired between the S and R terminals of the flip-flop, so those terminals are always out of phase, and the input is applied via a single pin. Fig. 5-*b* and Fig. 5-*c* show the symbol and the truth table of the type-D flip-flop, respectively.

A type-D flip-flop can be used as a data latch by connecting it as shown in Fig. 6-*a*, or as a binary counter/divider by connecting it as shown in Fig. 6-*b*.

The JK flip-flop

Figure 7-*a* shows the basic circuit of an even more versatile clocked flip-flop, which is universally known as the JK-type. It can function either as a data latch, a counter/divider, or as a do-nothing element by suitably connecting the J and K terminals. The symbol of the JK flip-flop is shown in Fig. 7-*b*, and its truth table is shown in Fig. 7-*c*.

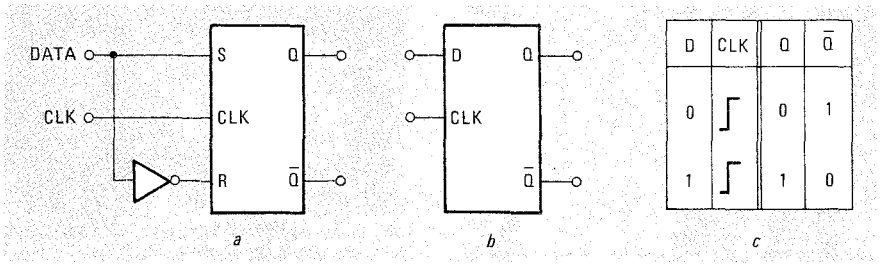
In essence, the JK flip-flop functions as a T-type when inputs are both high, and as a D-type when they're different. When they're both low, the outputs remain unchanged when a pulse arrives.

Real-world devices

The two best-known clocked CMOS flip-flops are the 4013 D-type and the 4027 JK-type. Each IC contains two independent flip-flops that share power and ground connections. Figure 8-*a* shows the functional diagram of the 4013; the truth table of its clocked inputs is shown in Fig. 8-*b*, and that of its direct inputs is shown in Fig. 8-*c*. Corresponding diagrams for the 4027 are shown in Fig. 9-*a*, Fig. 9-*b*, and Fig. 9-*c*.

Note that both the 4013 and the 4027 have SET and RESET inputs in addition to the normal clocked inputs. For both IC's those terminals are direct inputs that enable the clocked action of the flip-flop to be overridden, in which case the device functions as a simple unclocked S-R flip-flop. For normal clocked operation, the direct inputs must be grounded.

The 4013 and 4027 are fast-acting, so it is important that their clock signals be absolutely noise-free and bounceless, and that they have risetimes and falltimes of less than five μs. Both IC's clock on the positive transition of the clock signal.



D	CLK	Q	Q̄
0	⌐⌐	0	1
1	⌐⌐	1	0

FIG. 5—THE DATA OR TYPE-D FLIP-FLOP *a* is built from a clocked master-slave flip-flop. Its symbol is shown in *b*, and its truth table in *c*.

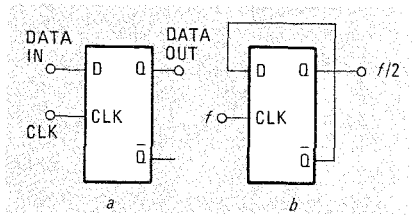


FIG. 6—THE D FLIP-FLOP can be used as a data latch (*a*) or as a divide-by-two counter (*c*).

Ripple counters

The most popular application of the clocked flip-flop is as a binary counter. Fig. 10-*a* shows how to connect the 4013 as a divide-by-two counter; Fig. 10-*b* shows the corresponding connections for the 4027. When clocked by a fixed-frequency waveform, both circuits give a symmetrical square-wave output at half the clock frequency.

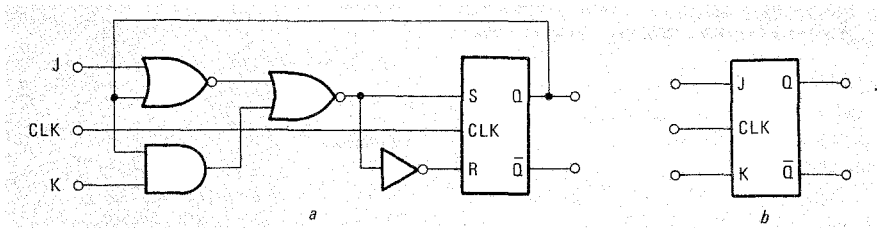
As shown in Fig. 11, you can cascade several ripple counters (so called because of the way that clock pulses appear to ripple from stage to stage) to provide division by successive powers of two. Figure 11-*a* shows how to cascade two D-type flip-flops, and Fig. 11-*b* shows how to cascade two JK-type flip-flops to provide a division ratio of 4 (2×2 or 2^2). In a like manner, Fig. 12-*a* and Fig. 12-*b* show how three stages can be cascaded to give a division ratio of eight (2^3). In fact, an arbitrary number of stages can be cascaded, as shown in Fig. 13, to provide a division ratio of 2^n, where n is the number of stages.

The circuits shown in Fig. 11–Fig. 13 are known as ripple counters, because each stage is clocked by the output of the preceding stage, rather than by a master clock signal. The effect, therefore, is that the clock signal seems to "ripple" through the counter chain. The problem is that the propagation delays of all the dividers add together and provide a delay that prevents the counter stages from clocking synchronously. Counters of that sort are in fact called asynchronous counters. If the outputs of the stages are decoded via gate networks, output glitches and inaccurate decoding can result.

Long ripple counters

Although 4013 and 4027 counters can be cascaded to give any desired number of stages, when more than four stages are needed, it's usually economical to use a special-purpose MSI ripple-carry binary counter/divider IC. Our next few figures show several examples.

The 4024, shown in Fig. 14, is a seven-stage ripple counter; all seven outputs are externally accessible. The IC provides a maximum division ratio of 128 (2^7). The



J	K	CLOCKING ACTION
0	0	DOES NOTHING (INHIBITS)
0	1	SETS Q OUTPUT LOW
1	0	SETS Q OUTPUT HIGH
1	1	CHANGES OUTPUT STATE

c

FIG. 7—THE JK FLIP-FLOP'S CIRCUIT is shown in *a*, along with its symbol (*b*) and action table (*c*.)

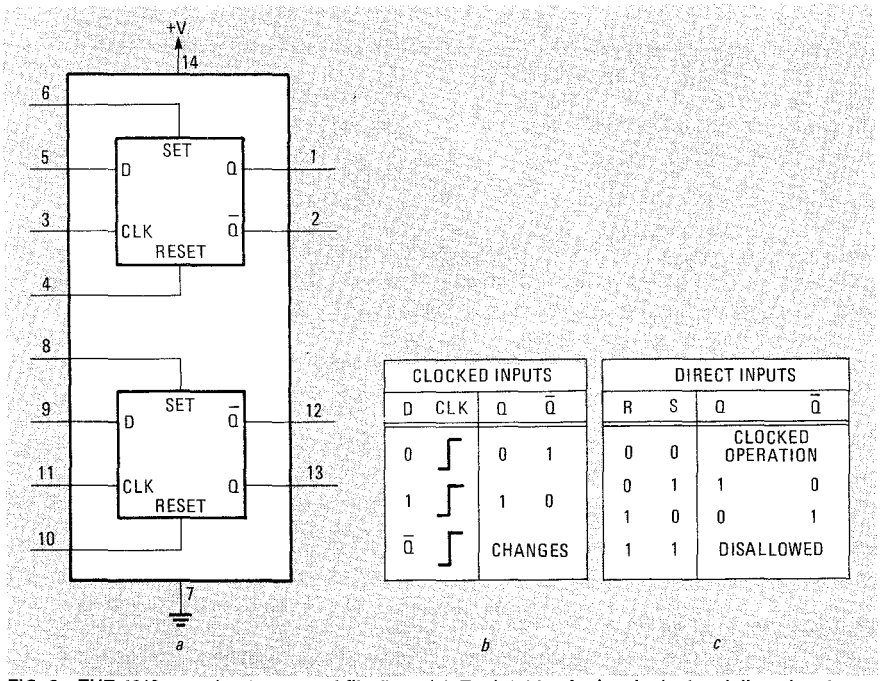


FIG. 8—THE 4013 contains two type-d flip-flops (*a*). Truth tables for its clocked and direct inputs are shown in *b* and *c*, respectively.

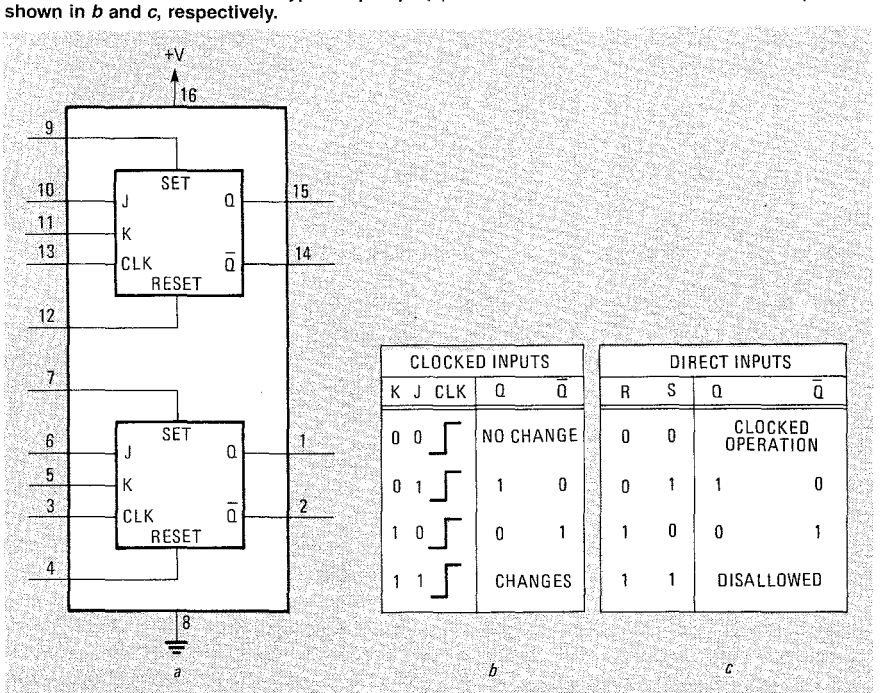


Fig. 9—THE 4027 contains two JK flip-flops (*a*). Truth tables for its clocked and direct inputs are shown in *b* and *c*, respectively.

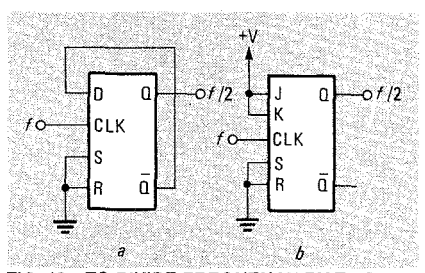


FIG. 10—TO DIVIDE FREQUENCY BY TWO, you can use a D (*a*) or a JK (*b*) flip-flop.

4040, shown in Fig. 15, is a 12-stage device, of which all outputs are accessible. It provides a maximum division ratio of 4096 (2^{12}). The 4020, shown in Fig. 16, is a 14-stage counter; all outputs except 2 and 3 are externally accessible. The 4020 provides a maximum division ratio of 16,384 (2^{14}).

Figure 17-*a* shows details of the 4060. It is another 14-stage device, but outputs 1, 2, 3, and 11 are not accessible. A special feature of the IC is that it incorporates a built-in oscillator circuit. As shown in Fig. 17-*b* and Fig. 17-*c*, the device can use either a crystal or an RC network to set the frequency of oscillation.

The 4020, 4024, 4040, and 4060 IC's all have Schmitt-trigger inputs that trigger on the negative transition of each input pulse. All of those counters can be set to zero by applying a high level to the RESET line.

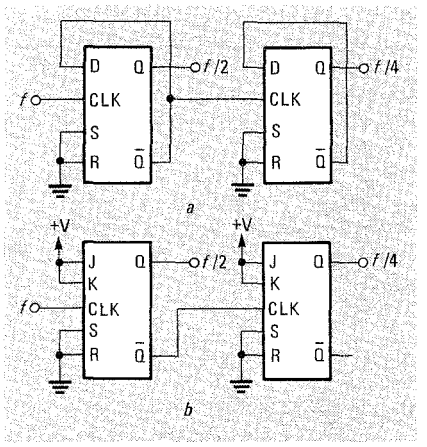


FIG. 11—TO DIVIDE FREQUENCY BY FOUR, you can use a pair of D (*a*) or JK (*b*) flip-flops.

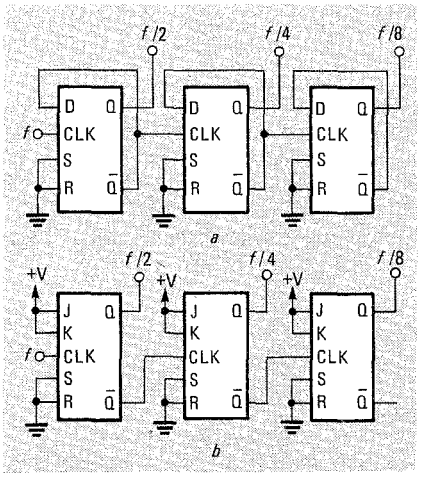


FIG. 12—TO DIVIDE FREQUENCY BY EIGHT, you can use three D (*a*) or JK (*b*) flip-flops.

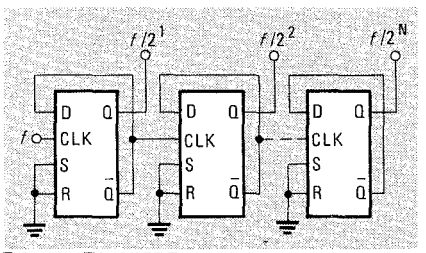


FIG. 13—TO DIVIDE FREQUENCY by an arbitrary factor 2^n, use *n* stages.

Glitches

A two-stage divide-by-four ripple counter, like that shown in Fig. 18-*a*, can have four possible output states, as shown in Fig. 18-*b*. Both outputs can be high, both can be low, one can be high and the other low, or the former low and the latter high. Before any clock pulses have been received, the Q2 and Q1 outputs are low. When the first pulse arrives, Q1 goes high. When the second pulse arrives, Q2 goes high and Q1 goes low. On the third pulse, Q2 and Q1 both go high. Last, on the fourth pulse, Q2 and Q1 both go low again.

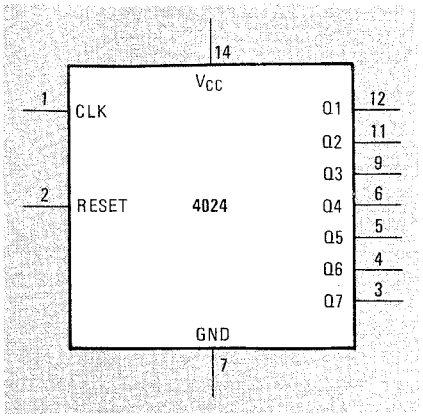


FIG. 14—PINOUT OF THE 4024 seven-stage ripple counter is shown here.

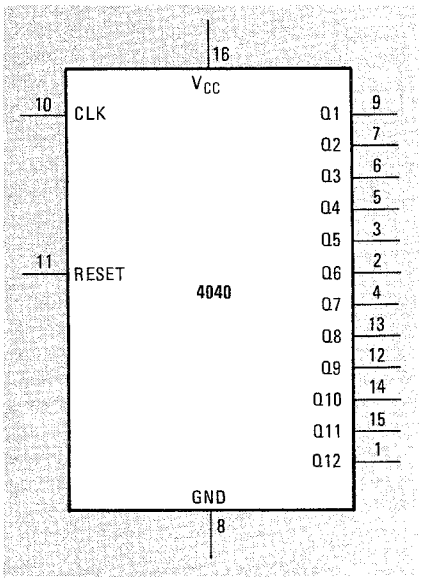


FIG. 15—PINOUT OF THE 4040 12-stage ripple counter is shown here.

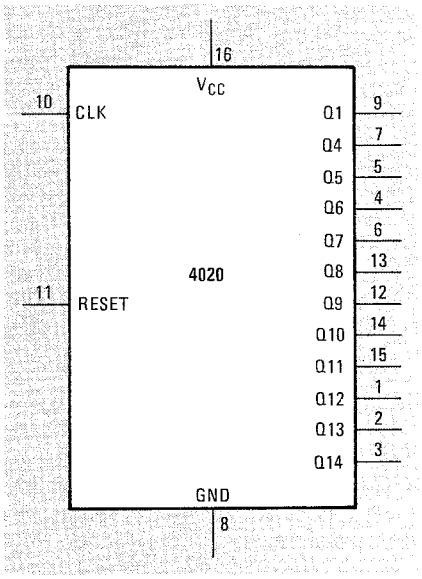


FIG. 16—PINOUT OF THE 4020 14-stage ripple counter is shown here.

Each of the four possible states can be decoded to provide four unique outputs by ANDing the outputs that are unique to each state, as shown in Fig. 18-*c*. Because the ripple counter is an asynchronous device, however, the propagation delay between the two flip-flops may cause glitches to appear in the decoded outputs, as shown in Fig. 18-*d*. Of course, those types of glitches are possible with any multi-stage ripple counter, and the greater the number of stages, the greater the total propagation delay becomes, and the greater the problem with glitches. The solution to the glitch problem is to use a clocked-logic device, which we'll discuss momentarily.

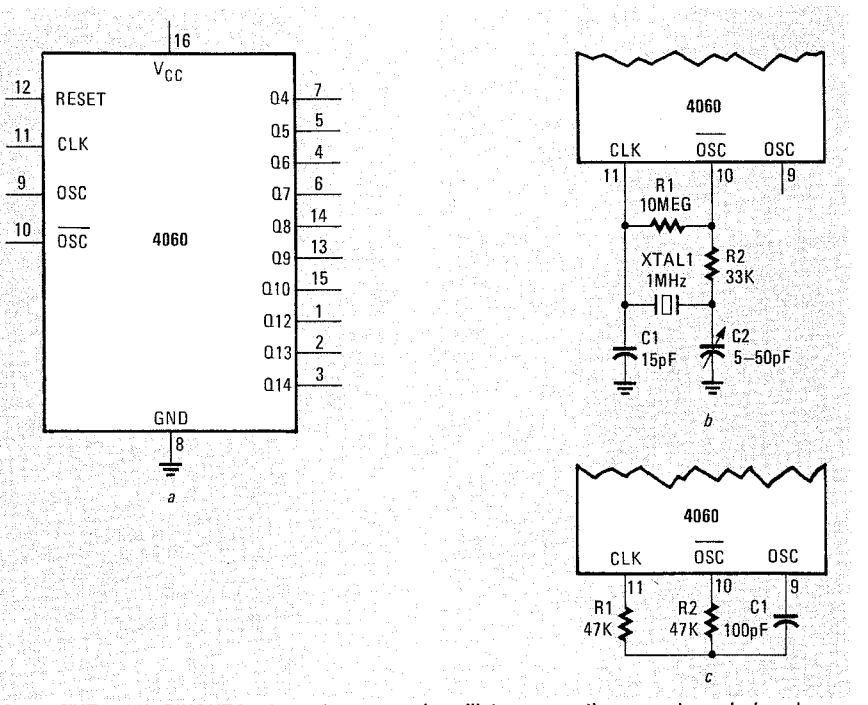


FIG. 17—THE 4060's PINOUT is shown in *a*; several oscillator connections are shown in *b* and *c*.

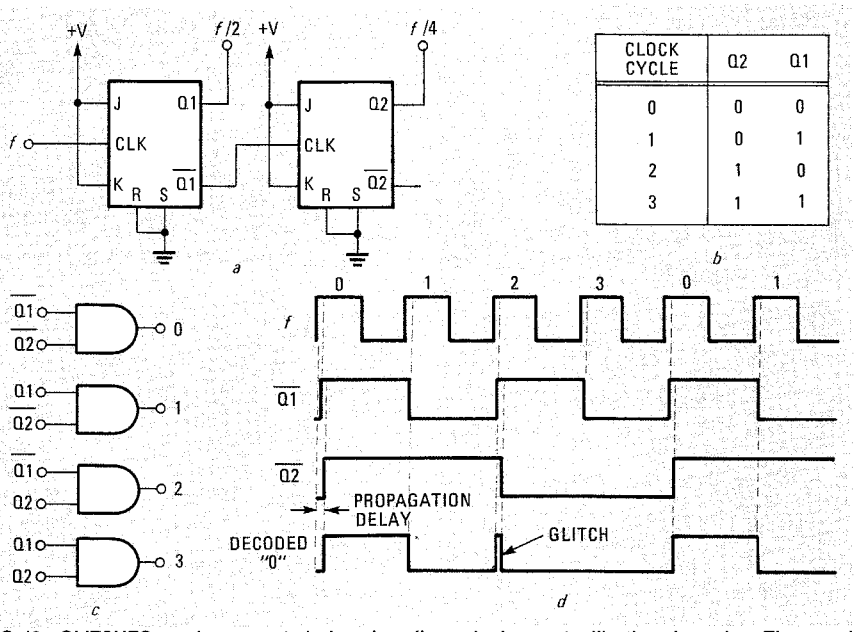


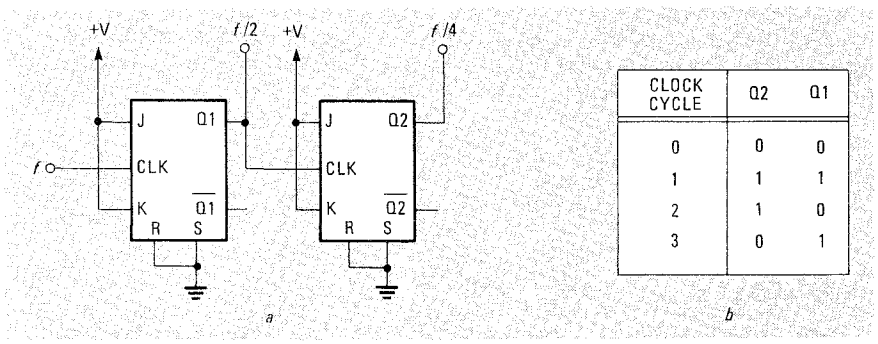
FIG. 18—GLITCHES may be generated when decoding a ripple counter like that shown in *a*. The Q_1 and Q_2 outputs respond to the input signal as shown in *b*. When they're combined as shown in *c*, a glitch may be generated, as shown in *d*.

Up and down counters.

A standard ripple counter counts up—the decoded outputs increase in value with each succeeding clock pulse. It is possible, however, to build a counter that works in the opposite direction. That type of counter is called a down (or a subtract) counter. The circuit is shown in Fig. 19-*a*; its truth table is shown in Fig. 19-*b*.

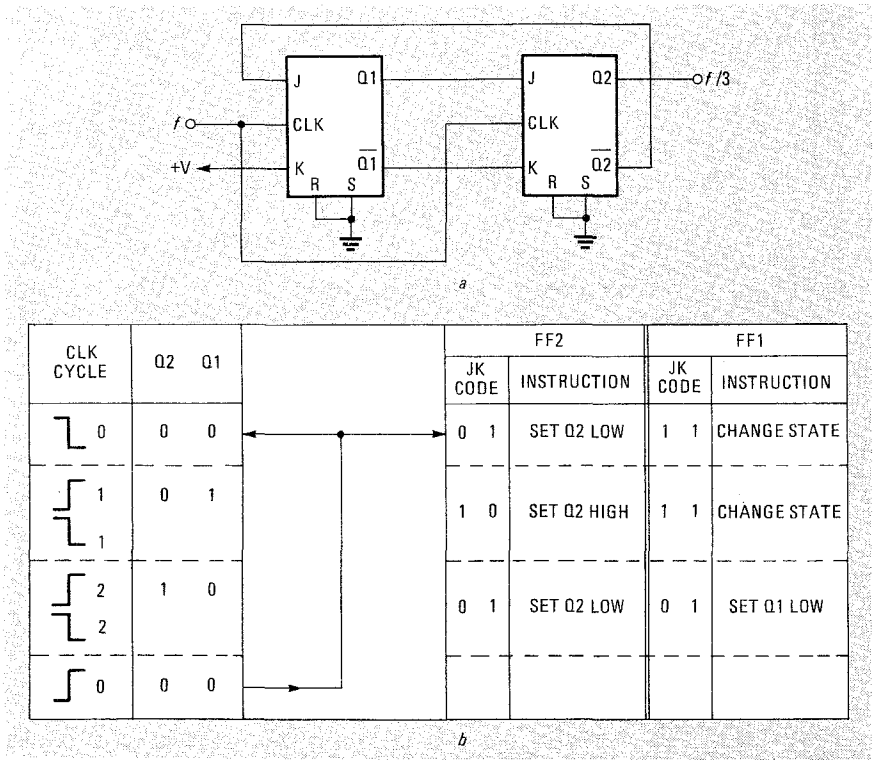
Walking-ring (Johnson) counters

Ripple counters are useful where undecoded binary division is needed, but



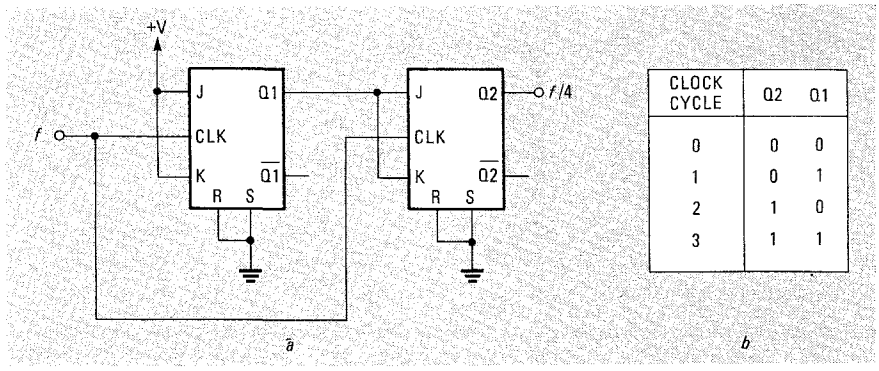
CLOCK CYCLE	Q2	Q1
0	0	0
1	1	1
2	1	0
3	0	1

FIG. 19—A DOWN-COUNTING RIPPLE COUNTER is shown in *a*; the truth-table is shown in *b*.



CLK CYCLE	Q2	Q1	FF2 JK CODE	FF2 INSTRUCTION	FF1 JK CODE	FF1 INSTRUCTION
0	0	0	0 1	SET Q2 LOW	1 1	CHANGE STATE
1	0	1	1 0	SET Q2 HIGH	1 1	CHANGE STATE
2	1	0	0 1	SET Q2 LOW	0 1	SET Q1 LOW
0	0	0				

FIG. 20—THE SYNCHRONOUS COUNTER eliminates glitches; a divide-by-three circuit is shown in *a*, and its truth-table in *b*.



CLOCK CYCLE	Q2	Q1
0	0	0
1	0	1
2	1	0
3	1	1

FIG. 21—A SYNCHRONOUS DIVIDE-BY-FOUR CIRCUIT is shown in *a*; its truth table is shown in *b*.

(because of glitches) not where decoded outputs are required. Fortunately, an alternative circuit, which is suitable for generating decoded outputs, is available. It is known as the walking ring or Johnson counter. It relies on the "programmable" nature of the JK flip-flop, which enables it to act as a SET (or a RESET) latch, as a binary divider, or as a "do nothing" device. In a walking-ring counter, all flip-flops are clocked simultaneously, so it is also known as a synchronous counter.

Figure 20-*a* shows the circuit and Fig. 20-*b* the truth table of a synchronous divide-by-three counter. Note that the truth table shows the state of each flip-flop at each stage of the counting cycle. Remember that, when the clock is low, the "instruction" is loaded (via the J and K inputs) into the flip-flop; the instruction is carried out as the clock goes high.

So, at the start of the cycle, Q2 and Q1 are both low, and the "change state" instruction (JK code 11) is loaded into the first flip-flop. Then the instruction "set Q2 low" (JK code Ø1) is loaded into the first flip-flop. When the first clock pulse arrives, the instruction is carried out, Q1 goes high, and Q2 stays low.

When the clock goes low again, new program information is fed to the flip-flops. Flip-flop 1 is instructed to change state (JK code 11), and flip-flop 2 is instructed to set Q2 high (JK code 1Ø). Those instructions are executed on the positive transition of the second clock pulse, causing Q2 to go high and Q1 to go low. When the clock goes low again, new program information is again fed to each flip-flop from the output of its partner. The counting sequence then repeats *ad infinitum*.

So in the walking-ring or Johnson counter, all flip-flops are clocked in parallel, but are cross-coupled so that the response of one stage (to a clock pulse) depends on the states of the other stages.

Walking-ring counters can be configured to give any desired count ratio. For example, Fig. 21-*a* and Fig. 21-*b* show the circuit and truth table respectively of a divide-by-four counter. Figure 22-*a* and Fig. 22-*b* show the circuit and truth table respectively of a divide-by-five counter.

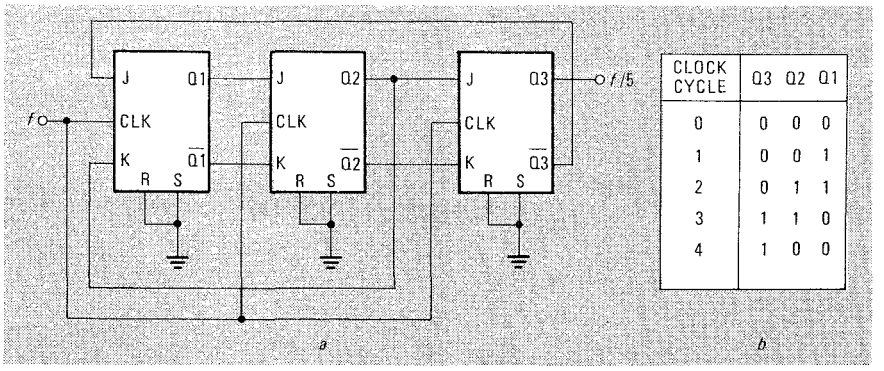
The 4018

When synchronous counts greater than four are needed, it is usually economical to use an MSI IC rather then several 4027's. A suitable device is a 4018, a presettable divide-by-N counter that can be made to divide any whole number between 2 and 10 by cross-coupling input and output terminals in various ways. That IC incorporates a five-stage Johnson counter, has a built-in Schmitt trigger in its clock line, and clocks on the positive transition of the input signal. The counter is said to be presettable because the outputs can be set to a desired state at any time by feeding the inverted binary code to the Jam inputs (J1–J5) and then loading the data by taking pin 10 high.

Figure 23 shows how to connect the 4018 to give any whole-number division ratio between 2 and 10. No additional components are needed to obtain an even division ratio, but a two-input AND gate (a 4081, for example) is required to obtain an odd division ratio.

Greater-than-ten division

Even division ratios greater than ten can usually be obtained simply by cascading suitably scaled counter stages, as shown in Fig. 24-*a*–Fig. 24-*d*. Non-standard and uneven division ratios can be



CLOCK CYCLE	Q3	Q2	Q1
0	0	0	0
1	0	0	1
2	0	1	1
3	1	1	0
4	1	0	0

FIG. 22—A SYNCHRONOUS DIVIDE-BY-FIVE CIRCUIT is shown in *a*; its truth table is shown in *b*.

Figure 26 shows how to make a four-bit *S*erial-*I*n/*S*erial-*O*ut (SISO) shift register. A bit of binary data applied to the input is passed to the output of the first flip-flop on the application of the first clock pulse, to the output of the second on the second pulse, to the output of the third on the third pulse, and to the fourth (and final) output on the fourth pulse. The circuit can hold four bits of data at any given moment. The SISO register is useful for delaying binary signals, or for storing bits of binary data and unloading them (in serial form) when required.

Figure 27 shows how the previous circuit can be converted to a *S*erial-*I*n/*P*arallel-*O*ut (SIPO) shift register simply by using the Q outputs of each flip-flop. The circuit might be useful, for example, in converting data transmitted from a remote location in serial form to the parallel form used by computers. **R-E**

DIVISION RATIO	FEEDBACK CONNECTIONS
2	$\overline{Q1}$ TO DATA
3	$\overline{Q1}$, $\overline{Q2}$ → TO DATA
4	$\overline{Q2}$ TO DATA
5	$\overline{Q2}$, $\overline{Q3}$ → TO DATA
6	$\overline{Q3}$ TO DATA
7	$\overline{Q3}$, $\overline{Q4}$ → TO DATA
8	$\overline{Q4}$ TO DATA
9	$\overline{Q4}$, $\overline{Q5}$ → TO DATA
10	$\overline{Q5}$ TO DATA

FIG. 23—TO OBTAIN AN ODD DIVISION RATIO with the 4018, an external AND gate must be used.

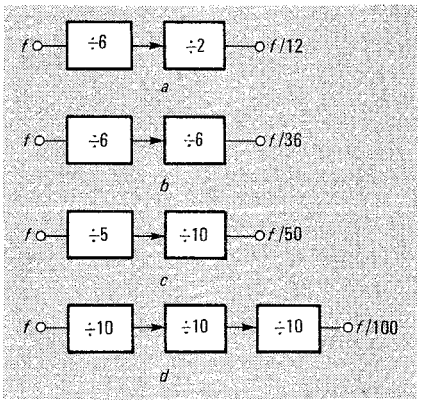


FIG. 24—A DIVISION RATIO OF arbitrarily large size may be obtained by cascading divider stages and multiplying the division factor. In *a* is a divide-by-12 *(2 × 6)* circuit, in *b* a divide-by-36 circuit *(6 × 6)*, in *c* a divide-by-50 *(5 × 10)*, and in *d* a divide-by-1000 *(10 × 10 × 10)* circuit.

obtained by using a standard synchronous counter (the 4018, for example) and decoding the outputs to generate suitable counter-reset pulses when the desired count is attained.

Latches and registers

Now let's move away from counters and take a brief look at three other applications of the clocked master-slave flip-flop.

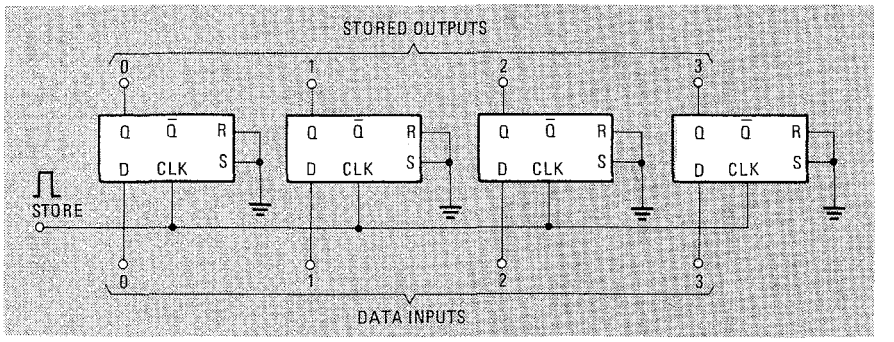


FIG. 25—TO STORE FOUR BITS OF DATA, all four inputs are clocked simultaneously.

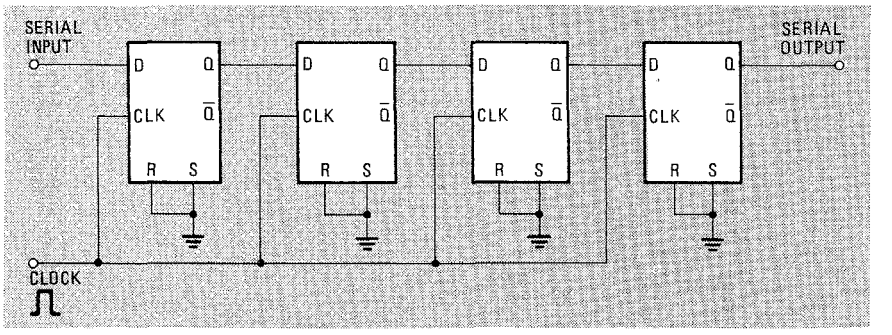


FIG. 26—FOUR TYPE-D FLIP-FLOPS are cascaded to create a four-bit SISO shift register.

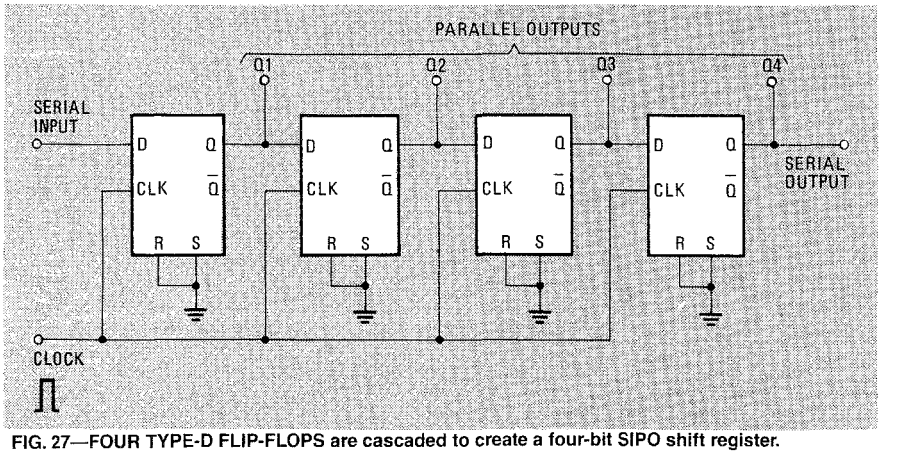


FIG. 27—FOUR TYPE-D FLIP-FLOPS are cascaded to create a four-bit SIPO shift register.

Figure 25 shows how to make a four-bit data latch from four D-type flip-flops. The data latch is useful for storing binary numbers or data. Input data is ignored until a positive-going STORE pulse is applied, at which point the latch stores the data and outputs it on the Q outputs.