SPECIAL COUNTER CIRCUITS FOR EXPERIMENTERS

Various combinations of the basic J-K flip-flop circuits provide many useful digital counting schemes

BY PATRICK J. DELANEY

WHEN the need for counting events in an electronic system arises, the most common solution is to use an asynchronous digital ripple counter. This circuit, shown in Fig. 1, consists of a group of J-K flip-flops set to toggle with each applied Clock pulse. The flip-flops are connected in cascade so that each Clock input is driven by the Q output of the preceding stage. The circuit provides a sequential binary-up count ranging from zero to one less than 2^N where N is the number of flip-flops in the circuit. While this arrangement finds wide use, it is only one of many useful digital counting schemes.

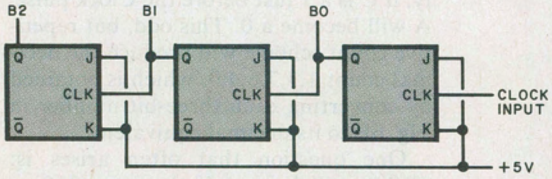
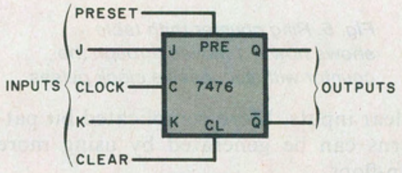


Fig. 1. Basic three-stage synchronous ripple counter.

Special counters make up a major class of flip-flop circuits. However, before these advanced configurations can be described in detail, a review of the J-K flip-flop is in order. A 7476 will be used as an example.

As shown in Fig. 2, a J-K flip-flop typically has five inputs and two outputs. The five inputs are Preset, Clear, Clock, J and K, while the two outputs are variously referred to as Q and not-Q, or some other alphabetic labels indicative of a variable and its complement. When a chain of flip-flops is used, the outputs are identified by a succession of alphabetic labels—A and not-A, B and not-B, etc.



INPUTS		OUTPUTS
PRESET	CLEAR	Q
0	0	Indeterminate (never used)
0	1	1 (SET)
1	0	0 (RESET)
1	1	Determined by J, K, and clock

Fig. 2. Details of the 7476 flip-flop and its truth table.

Preset and Clear inputs take precedence over all other inputs and are active low. This means that logic zero applied to the Preset input forces the Q output high (1). Alternately, Q would be forced low (0) by applying a logic zero to the Clear input. If both the Preset and Clear inputs are 1, the flip-flop's state will be determined by the J, K, and Clock inputs.

The last of the four possible combinations of these inputs is 0 for both Preset and Clear. This is contradictory in that it attempts to force Q high and low simultaneously (an impossible condition). The logic state of Q is thus indeterminate. The operation of the Preset and Clear inputs is summarized by the truth table shown in Fig. 2.

The J, K, and Clock inputs together form the second means by which the outputs may be changed. When these inputs are to be used, the Preset and Clear terminals are both tied high. When the J and K inputs are supplied signals as shown in Fig. 3, each time the Clock input is forced to switch from a logic 1 to a logic 0 (a negative edge), outputs take on the appropriate values. It should be noted that the J and K inputs alone have absolutely no effect on the flip-flop's output state. These two levels are clocked into the flip-flop by a one-to-zero transition at the Clock input. Figure 3 illustrates the effect of the four possible combinations of the J and K inputs on the Q output.

INPUTS			OUTPUT
CLOCK	J	K	Q
↓	0	0	Previous value
↓	0	1	0 (RESET)
↓	1	0	1 (SET)
↓	1	1	TOGGLE

Fig. 3. Effect of the J-K inputs on the Q output of the 7476.

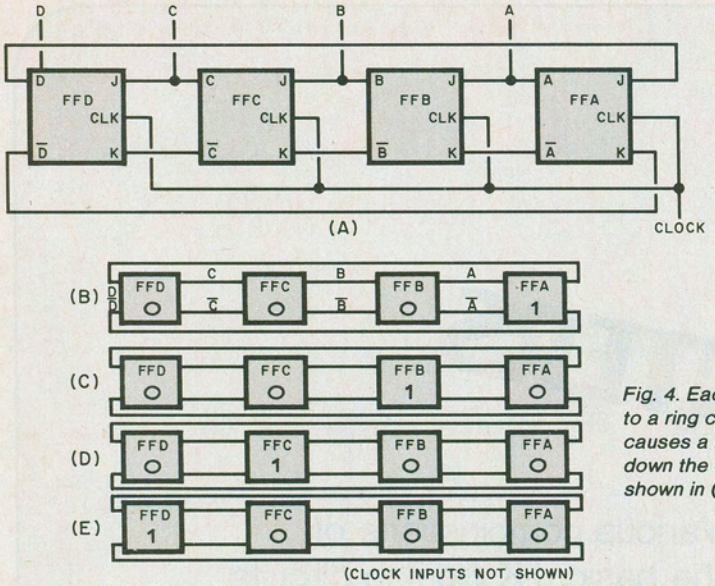


Fig. 4. Each clock pulse to a ring counter (A) causes a logic 1 to "walk" down the counter as shown in (B), (C), (D), and (E).

If both J and K equal 0, and a negative-going Clock pulse occurs, the Q output will remain unchanged. If J is equal to 1 and K equals 0 during the Clock edge, Q will become a logic 1 (set). When J equals 0 and K equals 1, a Clock pulse will force Q to equal 0 (reset). Finally, if J and K equal 1, the Clock will force Q to change its state or toggle. That is to say: if Q had been 0, it becomes 1 and if Q had been 1, it becomes 0.

Now that the operation of the J-K flip-flop is firmly within grasp, let us examine one of the special counters.

Ring Counter. A simple synchronous ring counter is shown in Fig. 4A. Note that the outputs of flip-flop D are fed back to the J and K inputs of flip-flop A, thus forming a "ring." The clock feeds all four clock inputs simultaneously. This circuit differs from most other counters in that its entire operation depends on the initial (power-up) states of its flip-flops. If, for example, output A equals 1 while outputs B, C and D equal 0 as in Fig. 4B, then the application of a single Clock pulse will set B, but Reset A, C and D because the J and K inputs of B were 1 and 0 (the condition for Set) the instant before the Clock pulse, but the J and K inputs of all of the other flip-flops were 0 and 1 respectively. The new condition is shown in Fig. 4C. The next Clock pulse will set C and reset A, B and D, as shown in Fig. 4D. Figure 4E shows how a third Clock pulse will Set D and Reset A, B and C. The final Clock pulse will cause A to set once again, and the cycle will repeat. Note that a lone 1 is shifted from right (A) to left (D) and then back around again. This changing pattern is responsible for the circuit's name. The 4-bit pattern shifted through the group of flip-flops, or register as they are collectively called, is not limited to a single 1, but can be any one of many patterns. These bit patterns can be forced into the register, before the application of Clock pulses, by the use of the Preset and Clear inputs. More complicated bit patterns can be generated by using more flip-flops.

CLOCK	D	C	B	A
1	0	0	0	1
2	0	0	1	0
3	0	1	0	0
4	1	0	0	0
5	0	0	0	1
6	0	0	1	0
7	0	1	0	0
8	1	0	0	0
9	0	0	0	1
10	0	0	1	0
11	0	1	0	0
12	1	0	0	0

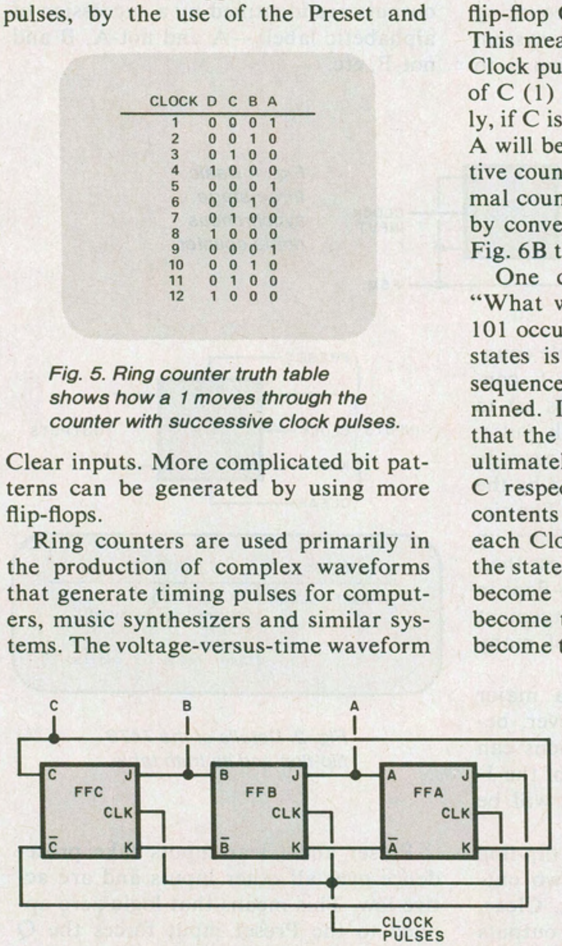
Fig. 5. Ring counter truth table shows how a 1 moves through the counter with successive clock pulses.

Ring counters are used primarily in the production of complex waveforms that generate timing pulses for computers, music synthesizers and similar systems. The voltage-versus-time waveform resulting from the truth table in Fig. 5 is easily obtained by simply turning the column on its side, and drawing a 5-volt waveform for each 1 and a zero-volt waveform for each 0. Note that the negative-going edge of the Clock triggers each transition of the outputs.

Shift Counters. A small change in the feedback from the last to the first flip-flop of a basic ring counter produces the shift counter shown in Fig. 6A. Unlike the ring counter, the shift counter is usually used as a synchronous event counter rather than as a waveform generator. Shift counters can easily produce any even modulus (number of states) count at extremely high rates of speed, with illegal states and high power consumption and component count being the only drawdacks. Another difference between the ring counter and the shift counter is that the latter does not have to be Preset to a specific pattern or starting state. The shift counter can naturally fall into the correct counting sequence.

A typical count sequence for a three-bit shift counter is shown in Fig. 6B. Whenver a Clock pulse occurs, the feedback connection between flip-flops C and A causes the inverse of the state of flip-flop C to be loaded into flip-flop A. This means that, when C = 0 before a Clock pulse, A will become the opposite of C (1) after the pulse ends. Conversely, if C is a 1 just before the Clock pulse, A will become a 0. This odd, but repetitive count scheme will produce the decimal count 1,3,7,6,4,0, which is obtained by converting each three-bit number in Fig. 6B to its decimal equivalent.

One question that often arises is: "What will happen if the state 010 or 101 occurs?" Since neither of these two states is part of the normal counting sequence, their effects should be determined. It has aready been established that the state of flip-flops A and B will ultimately be shifted to flip-flops B and C respectively, and that the inverted contents of C will be shifted into A after each Clock pulse. It is clear, then, that the state 010 (decimal 2) will force B to become the former state of A, C to become the former state of B, and A to become the inverse or opposite of C. It is



CLOCK	C	B	A	DECIMAL
↓	0	0	1	1
↓	0	1	1	3
↓	1	1	1	7
↓	1	1	0	6
↓	1	0	0	4
↓	0	0	0	0
↓	0	0	1	1

Sequence begins again

Fig. 6. Note the feedback from the last flip-flop to the first in a shift counter (left) with the truth table shown at right.

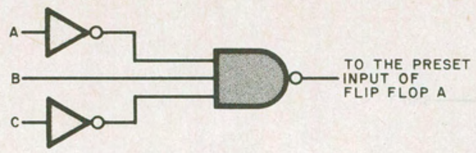


Fig. 7. Illegal state detector works in conjunction with the ring counter to avoid possible oscillation between 010 (decimal 2) and 101 (decimal 5).

as if all of the bits have shifted left by one position, while the most significant bit became complemented and replaced bit A. Thus, illegal state 010 (decimal 2) gives way to state 101 (decimal 5), while state 101 (decimal 5) gives way to the original state 010 (decimal 2). This means that either state, once entered, will cause an oscillation back and forth from decimal 2 to decimal 5, and the normal six-state count sequence will never be entered.

Since the initial application of power to a series of flip-flops creates a random state, it is possible that one of the two undesirable, illegal states will be entered at power-up. To prevent the 2-5-2-5 oscillation from persisting, a series of gates must be added to the basic shift counter. The purpose of this gating arrangement is to detect one of the illegal states and force legal counting to resume. Once the legal counting sequence begins, it continues to exclude the two undesired states.

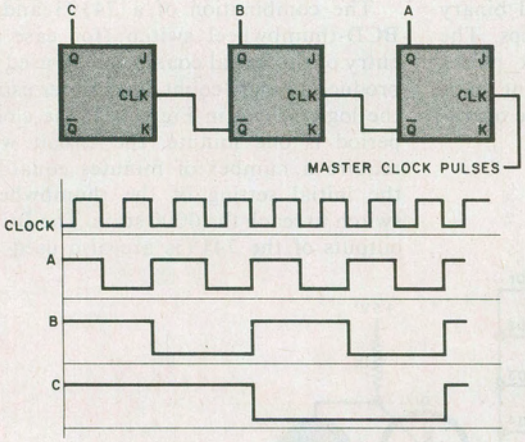
Logic state 010 can be detected by the arrangement shown in Fig. 7. The output of the NAND gate will become zero only when state 010 occurs at inputs A, B and C. This low level will instantly Preset flip-flop A, thus causing the state 011, or decimal-3 to be entered. From this point on, the shift counter will operate only in its legal six-state sequence.

Just as with the ring counter, a shift counter may employ any number of flip-flops. One or more count sequences containing a number of states equal to twice the number of flip-flops will always occur. Illegal states will also frequently exist and must be prohibited from occurring through the use of circuits similar to that of Fig. 7. It is left to the reader as an experiment to prove that a four-bit shift counter would produce two eight-bit sequences, one of which would generate very odd waveforms indeed!

Up/Down Counters. The last type of circuit falling into the category of special counters is the up/down counter. In its simplest form, the up/down counter is a variation of the basic ripple counter. In this circuit, each clock pulse causes a group of outputs to take on a binary value one larger than that existing previously. The primary reason for the ever-increasing count is the connection from the Q output of each flip-flop to the clock input of the next flip-flop. If the not-Q output of each flip-flop were used for this purpose, the count would decrease rather than increase. Occasionally, of course, the count would reach zero, at which point it would begin to decrease from the highest count again.

The logic and truth table for a down count are shown in Fig. 8. Note that all the flip-flops are wired so as to toggle with each input clock pulse. The toggle occurs on the negative-going trailing edge of the transition from 1 to zero. For example, the not-Q output of flip-flop A acts as a clock pulse for flip-flop B. Flip-flop B will not toggle (change state) each time the master clock input produces a negative edge, but rather when the not-Q output of flip-flop A switches from 1 to zero. In much the same fashion, flip-flop C responds only to the negative clock edges provided by the output of flip-flop B. The resulting count can clearly be seen to decrease rather than increase in the manner of a standard up ripple counter. A counting circuit that counts solely up or down can often be useful, but a circuit with the ability to count both up and down would in general, be even more useful. The previous discussion indicates that any ripple counter can be changed from count up to count down through simple rewiring.

However, true versatility can be achieved only by electronic control of the counting direction, using a circuit such as that shown in Fig. 9. This circuit makes use of the gating, or switching property of AND gates E and F. That property can best be summarized as follows whenever one input to the AND gate is a logic 1, the gate is said to be enabled, and the remaining gate input will be switched through to the output as if the gate were a piece of conducting wire. The inverter at the count up/down input will cause only the E pair of AND



CLOCK	C	B	A
1	1	1	1
2	1	1	0
3	1	0	1
4	1	0	0
5	0	1	1
6	0	1	0
7	0	0	1
8	0	0	0

Fig. 8. Three-stage down counter, its truth table and waveforms illustrating how counting takes place on the negative-going edge of the clock pulse.

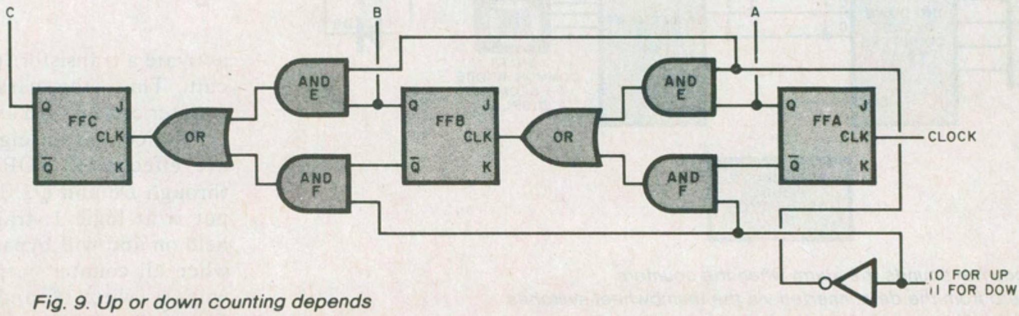


Fig. 9. Up or down counting depends on the signal applied to the inverter.

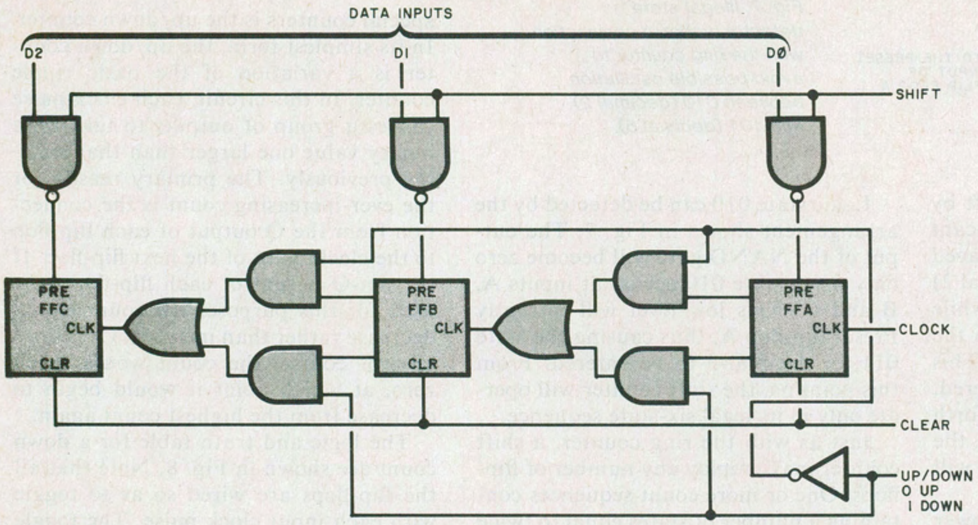


Fig. 10. One method of parallel data entry into an up/down counter. Data input can come from thumwheel or fixed switches, other digital logic, or even from a computer.

gates or the F pair of AND gates (but not both) to be enabled at any given instant. If the E pair is enabled, the Q output of each flip-flop serves as the clock input for the next flip-flop, since the F input to each OR gate will always be at logic 0, and the OR gate output will simply take on the value of the previous flip-flop Q output. This coupling of A to B clock and B to C clock will result in an up count.

If the count input is set to logic 1, the F-pair of AND gates will be enabled and each OR gate output will take on the value of the previous not-Q output. This is electrically similar to the circuit of Fig. 8. A down count will result because A is effectively gated to B clock, while B is effectively gated to C clock.

If the capability to load a given count into the up/down counter is provided, the utility of the circuit will be greatly enhanced. One approach which could be used to achieve parallel entry of data into an up/down counter is shown in Fig. 10. The application of a logic 0 to the normally high clear (CLR) line forces the Q output of all four flip-flops to a logic 0. This would be followed by a brief logic-1 shift pulse to load binary 1's into the appropriate flip-flops. The sequential application of clock pulses will now force the circuit to count up or down from this initial value. The operation of this circuit can essentially be duplicated by a conventional TTL 74193 IC with two small exceptions. The 74193 is an up/down decade counter and, as such, will count only to 1001 (decimal 9) before resetting to 0000. The second minor difference between the circuit of Fig. 10 and the 74193 is the method of parallel data entry. The IC abandons the cumbersome, two-step Clear-Shift approach for a simple one-pulse load technique.

The combination of a 74193 and a BCD-thumbwheel switch (for ease of entry of the initial count) can be used to produce a handy count down timer using the logic shown in Fig. 11. If the clock period is one minute, the circuit will require a number of minutes equal to the initial setting of the thumbwheel switch to reach the 0000 state. The BCD outputs of the 74193s are also used to activate a transistor (*Q1*) and alarm circuit. Thus, the parallel-loading down counter can be used as a presettable timer. Notice that the eight counter outputs are effectively NORed by diodes *D1* through *D8* and *Q1*. If any counter output is at logic 1, transistor *Q1* will be held on and will bypass the alarm. Only when all counter outputs are at logic 0 will *Q1* switch off and permit current to flow to the alarm. ◇

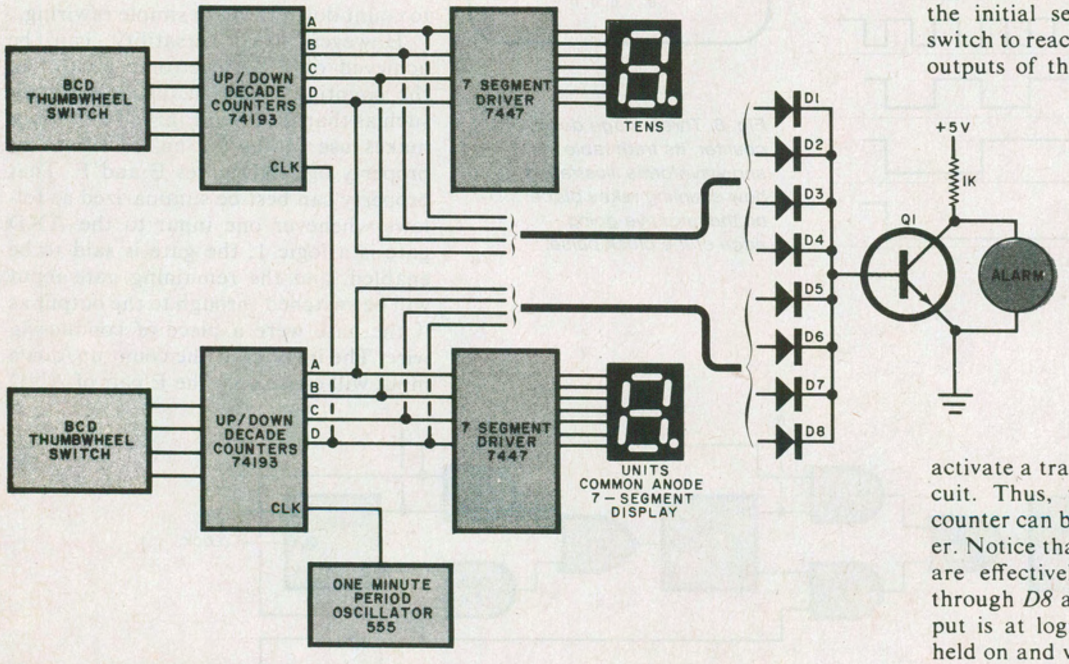


Fig. 11. Down counter sounds the alarm when the counters decrement to zero from the data inserted via the thumbwheel switches. In this case, there will be one decrement per minute.