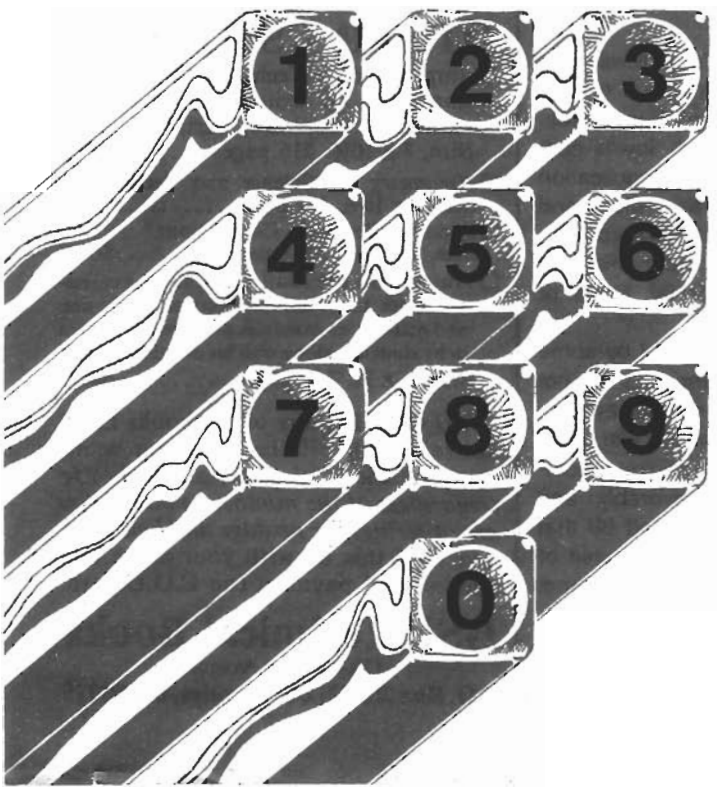
BY CLEMENT S. PEPPER

THE VERSATILE KEYPAD

How to interface these important mechanical devices with digital circuits.



THERE ARE an ever-increasing number and variety of low-cost decimal and hexidecimal keypads available to the electronics experimenter. To successfully use these keypads, one must observe certain criteria to be sure mutually compatible signals are available. You cannot just connect any keypad to any circuit and expect the system to operate properly. Either the keypad selected must be specifically designed for the digital circuit it is to drive, or the digital circuit must be designed to suit the specific keypad.

One major problem with keypads (and most other mechanical switches) is that they are not ideal switches. Instead of producing a single pulse when they are opened and closed, they produce a "train" of brief pulses as they mechanically settle. In ordinary switching applications, this "bouncing" is not a problem. But when switches are used with high-speed electronic counters, each pulse within a train (Fig. 1) can appear as a separate toggle signal, resulting in false counting.

Most keypads are decimal (0 to 9), while many electronic circuits require a binary-coded-decimal (BCD) input. Hence, a decimal-to-binary decoding system to make the conversion is required. Too, many counting circuits also require a "start" or "sync" signal to "tell" them when a key has been depressed. Therefore, some kind of key-closure sensing system must be used.

Debouncing. A basic debouncing circuit for a switch is shown in Fig. 2, accompanied by its truth table. The circuit consists of an AND and an OR gate. When the switch is closed, input A goes low and forces the output of the AND gate low. This low signal is connected to the C input of the OR gate and is additionally used to toggle the bounce-inhibit monostable multivibrator. In response to the low at its input, the multivibrator sends a low signal to the D input of the OR gate for a period of time determined by the monostable time constant. Since both inputs to the OR gate are low, the output of the gate also goes low.

The switch can now be released, causing the A input to go high, due to the pull-up resistor. With the low output of the OR gate connected to the B input, the output of the AND gate remains low. The circuit will remain in this state until the monostable time constant times out and sends a high signal to the D input of the OR gate.

As explained above, the very first closure of the switch causes the circuit to operate but locks out any subsequent bounce-produced signals. The only thing to keep in mind is that the bounce-inhibit monostable time constant must produce an output slightly longer than any expected bounce interval.

The circuit shown in Fig. 3 illustrates the use of the debounce circuit with a BCD coding scheme. A function truth table is also shown. You may be surprised to see a hexidecimal table for a 10-key array. If you wish to obtain a hex A (10),

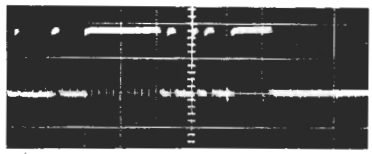
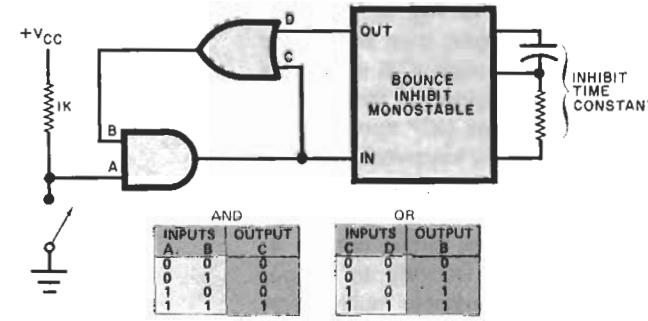


Fig. 1. Pulse train resulting from switch contact bounce. Sweep time is 50 μs/div.

both the 8 and 2 keys must be pressed simultaneously. Similarly, a hex F (15) requires simultaneous operation of the 8 and 7 keys. If you plan to use a hex keypad, use the same AND-OR gate logic for all 16 switches and substitute the circuit shown in Fig. 4.



AND

INPUTS A	INPUTS B	OUTPUT C
0	0	0
0	1	0
1	0	0
1	1	1

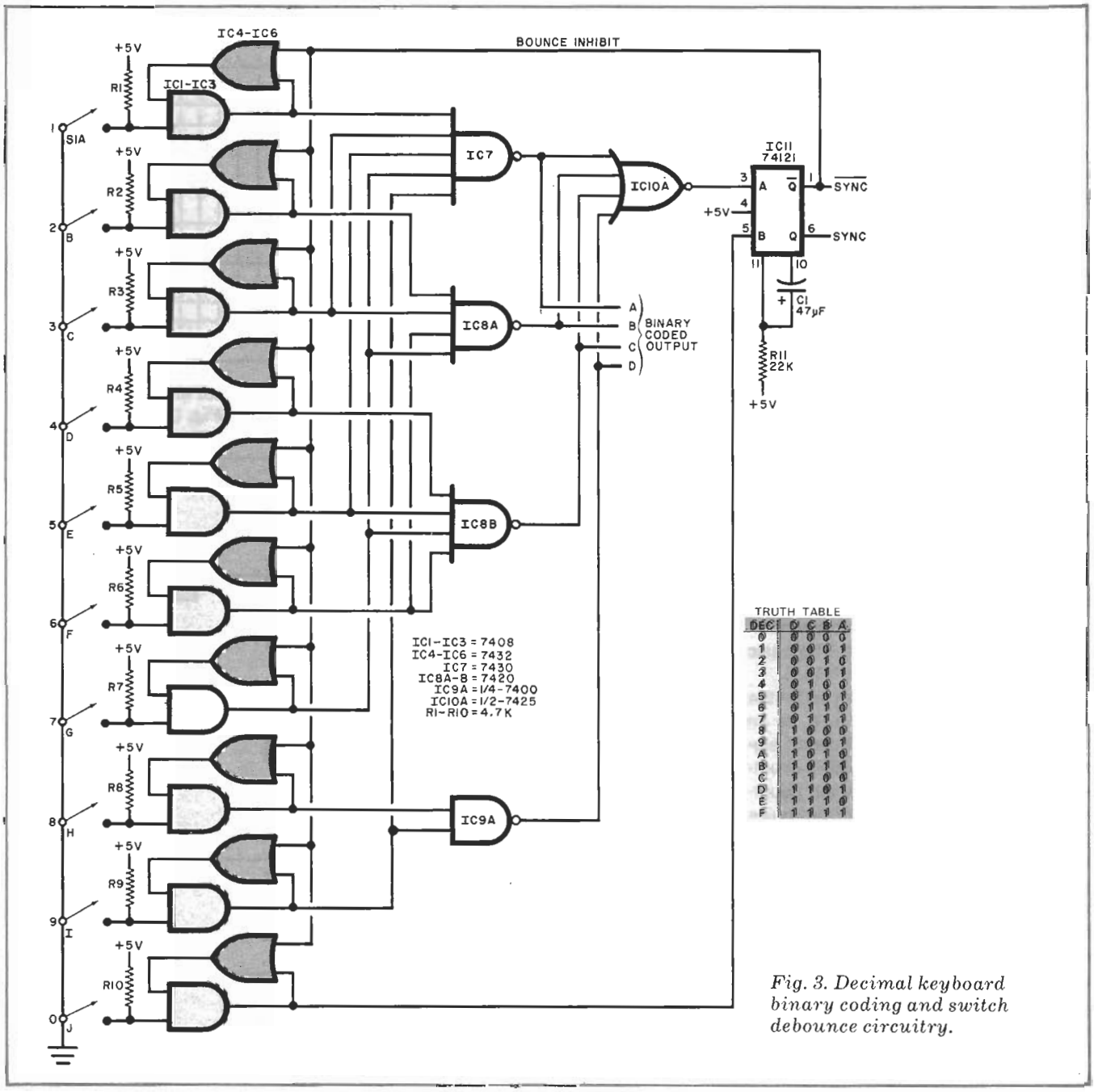
OR

INPUTS C	INPUTS D	OUTPUT B
0	0	0
0	1	1
1	0	1
1	1	1

Fig. 2. Switch debounce circuit is formed from AND-OR gate logic.

SWITCH CIRCUIT LOGIC

STATE	A	B	C	D	
0	1	1	1	1	Switch open
1	0	1	0	1	Switch closure
2	0	0	0	0	Debouncer response
3	1	0	0	0	Switch bounce
0	1	1	1	1	Switch open, Debounce reset



TRUTH TABLE

DEC	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
A	1	0	1	0
B	1	0	1	1
C	1	1	0	0
D	1	1	0	1
E	1	1	1	0
F	1	1	1	1

Fig. 3. Decimal keyboard binary coding and switch debounce circuitry.

Referring back to Fig. 3, when all keyswitches are open, their associated AND gate (*IC1* through *IC3*) inputs are high. Hence, the outputs of the four encoding NAND gates (*IC7* through *IC9*) are low. Closing any keyswitch except 0 forces at least one of the NAND gate inputs high.

The bounce-inhibit circuit uses a 4-input NOR gate (*IC10A*) to trigger bounce-inhibit monostable multivibrator *IC11*. When any of the four NOR gate inputs go high (any key closed), the output of the NOR gate goes low and triggers the multivibrator. The multivibrator, in turn, sends a low signal to the OR gate associated with each key. This implements the debounce function. For the RC values given in Fig. 3, the debounce period is about 700 ms. For the 74121 monostable multivibrator, the timing equation is T = 0.69RC, with R kept at a value of less than 40,000 ohms.

The circuit remains in the debounce condition and ignores any switch bounce until the monostable multivibrator times out. When this occurs, the circuit resets back to where another key can be operated. Note in Fig. 3 that the multivibrator also produces a "sync" signal in exact time step with the input pulse. This is for use with an external counting or other enabling circuit.

The 0 key requires a different approach from that discussed. Although it has the same debounce circuit as the other keys, when the 0 key is closed, a separate input trigger, B, on the multivibrator is used.

Controlled Pulse Generator. One use for a debounced and BCD-coded keypad is as a controlled pulse generator that delivers a number of output pulses determined by the decimal number inserted via the keypad. The basic logic for this circuit is shown in Fig. 5.

Pressing any key on the keypad in the Fig. 5 circuit sends a sync pulse to an enabling latch and the BCD-coded signal to the inputs of a binary down counter. The latch signal enables the counter's preset input and a controlled-pulse generator. The pulse generator is designed so that both pulse width and pulse period can be controlled. Each time a pulse appears at the ouput, the binary down counter is decremented by one. When the counter reaches zero, it resets the latch and stops the operation.

The actual circuit, shown in Fig. 6, is straightforward. The *IC1A/IC1B* latch is made from conventional TTL NAND gates, with RC coupling at the inputs to

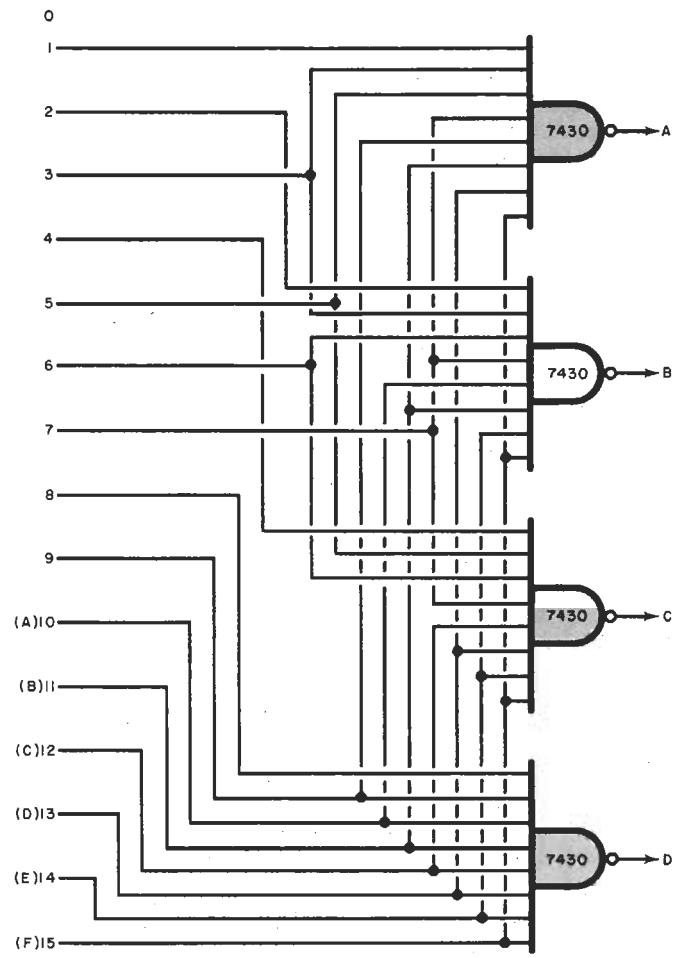


Fig. 4. Decoding logic for a hexidecimal keypad. This circuit is an addition to that in Fig. 3.

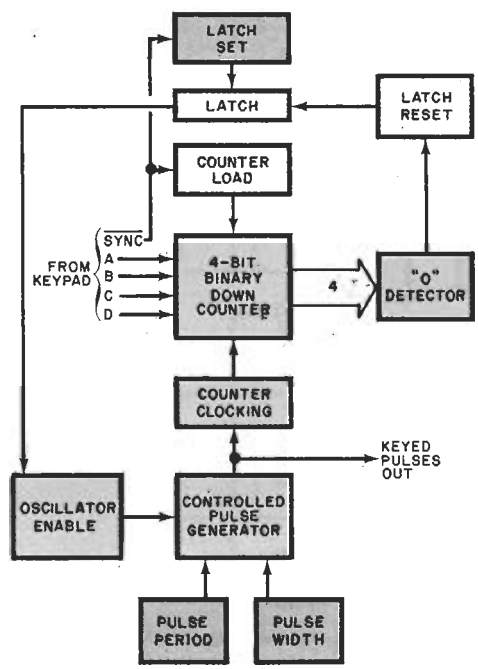


Fig. 5. Function diagram for a controlled-pulse generator.

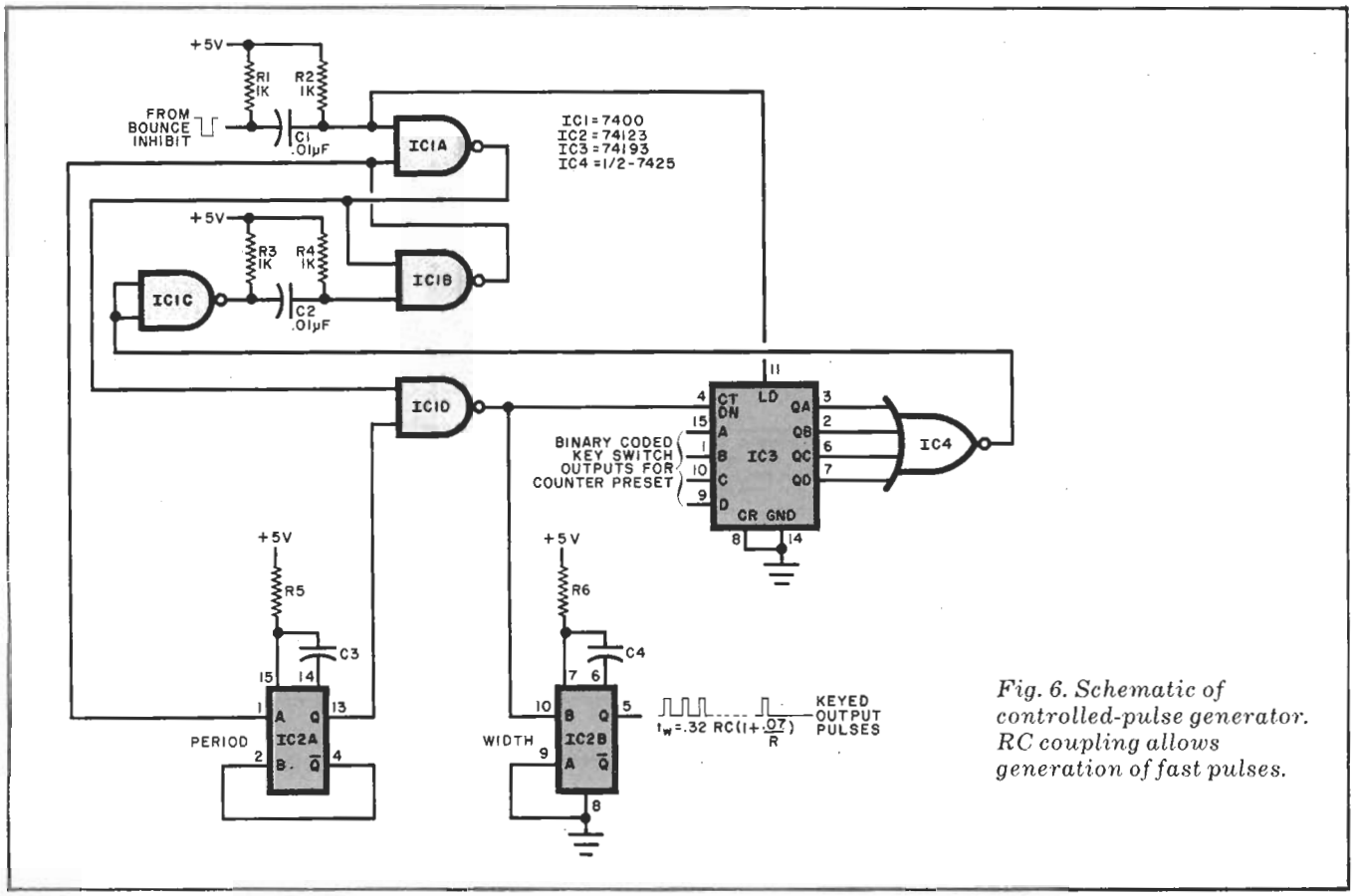


Fig. 6. Schematic of controlled-pulse generator. RC coupling allows generation of fast pulses.

allow rapid action—in fact, a complete pulse train can be generated within the width of the sync pulse. Without RC coupling, the latch would be locked for the duration of the sync time. A transient input is a must to avoid lockout. The *IC3* down counter has its LOAD enable input RC coupled to the sync input. This input requires a transient input to operate.

The controlled-pulse generator (*IC2*) is made up of both halves of a 74123 dual monostable multivibrator. The RC timing of *IC2A* sets the pulse period. The Q output at pin 13 is connected to NAND gate *IC1D*, with the second input of this gate connected to the latch. With the latch reset, the NAND gate is locked and its output remains in the high state, regardless of what the multivibrator is doing. In reality, *IC2A* is not doing anything, since its A input trigger at pin 1 is also enabled by the latch.

The first cycle of the operation is initiated when the latch is set. This causes a high-to-low transition at the A input. When the multivibrator triggers, the Q output at pin 4 goes low. When the multivibrator times out, the low-to-high transition at the Q output retriggers the multivibrator. Because the transition is so fast, the multivibrator appears to be continuously in the triggered state.

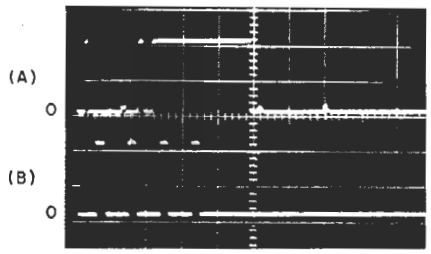


Fig. 7. Scope trace (A) shows switch bounce, while (B) shows four pulses initiated by switch closure. Sweep time is 50 μs/div.

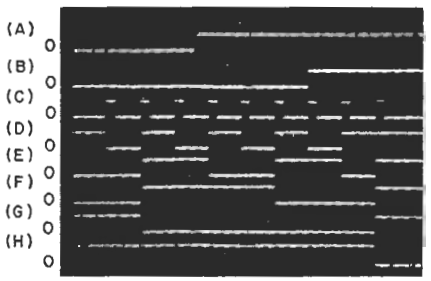


Fig. 8. Nine pulses generated by key switch closure (50 ms/div): (A) key closure; (B) sync; (C) outputs of 74123; (D) output QA; (E) output QB; (F) output QC; (G) output QD; all of IC3; and (H) latch input to IC1D.

The output of gate *IC1D* decrements the *IC3* counter and triggers the second monostable multivibrator (*IC2B*). The timing of this circuit controls the width of the pulse.

The only limitation on the frequency and width of the keyed pulses are those determined by the multivibrators. Very long and very short pulses over almost any range can be generated once the counter is preset. The keypad plays no role in this part of the operation.

The oscilloscope waveforms for the Fig. 6 circuit are shown in Fig. 7. The upper trace shows switch contact bounce, while the lower trace shows four pulses initiated by the first switch closure. Note the immunity to switch noise and the fast response possible. The traces in Fig. 8 show the timing of those functions that will be helpful in understanding the operation of the circuit.

Combination Lock. The logic for a four-digit combination lock that can be operated only by someone who knows the code is shown in Fig. 9. This circuit can easily be expanded so that several functions can be derived from a single keypad. Appropriate interfacing must be

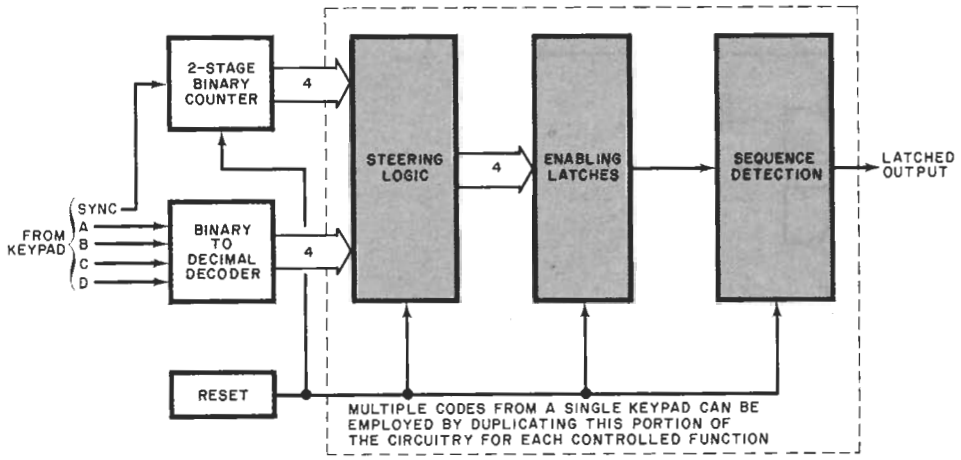


Fig. 9. Four-digit combination lock that works with only one selected set of input digits.

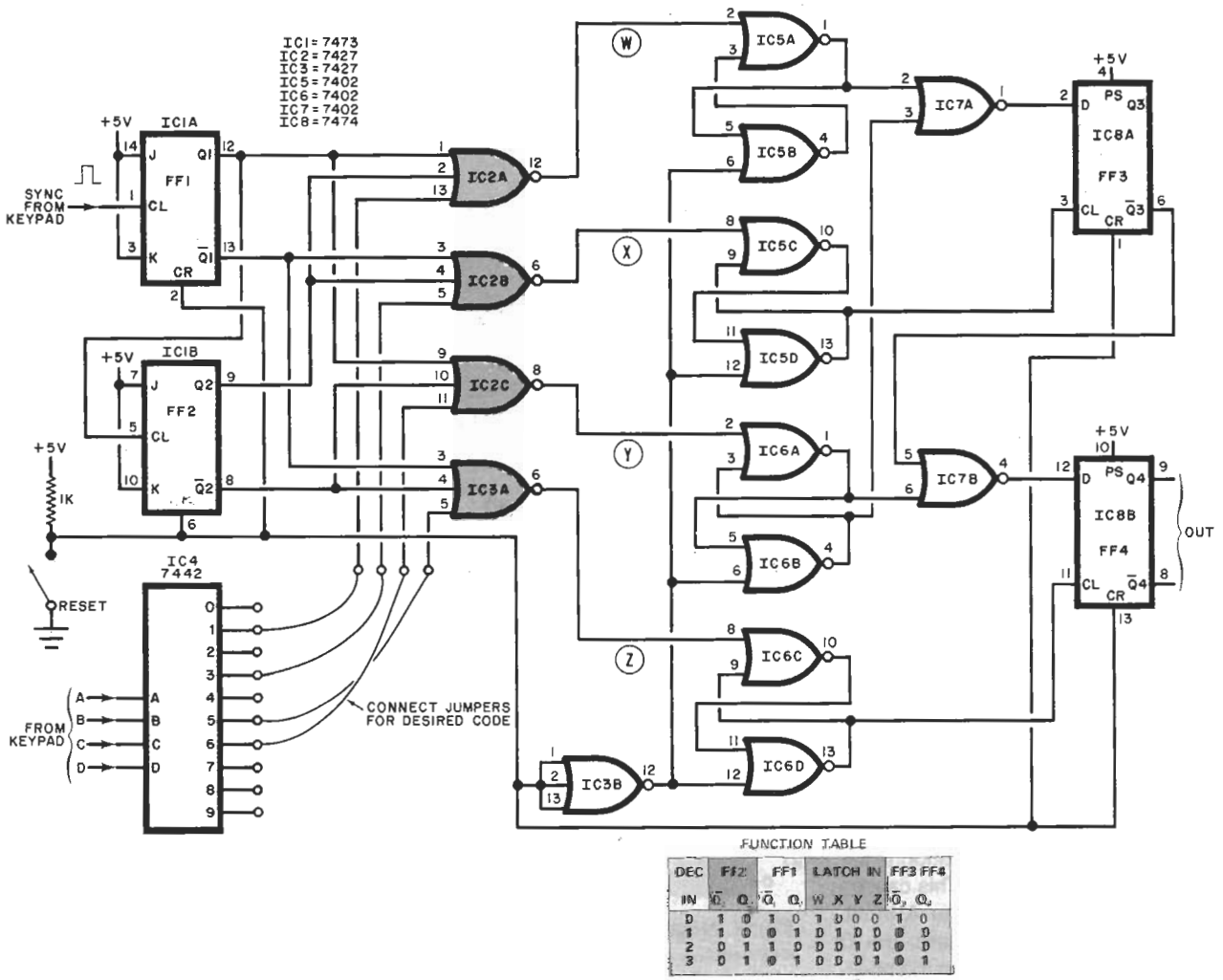


Fig. 10. Four-digit lock with combination 1365. Keyed code must match jumpered connections to operate lock.

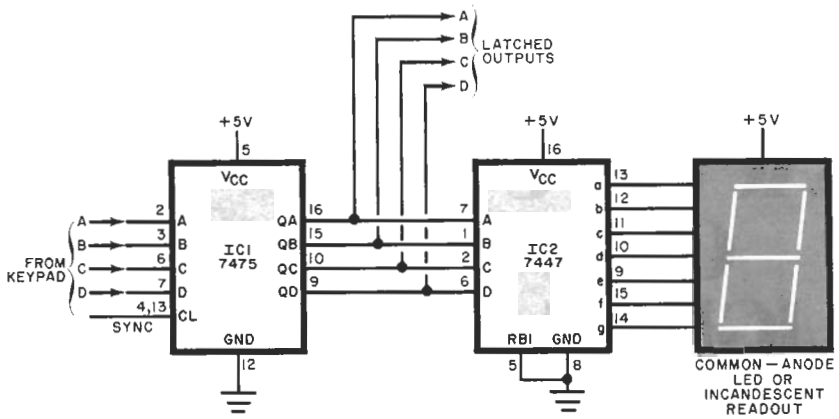


Fig. 11. Latched output for a keypad. Display is on a 7-segment LED readout.

added between the circuit and any external devices to be controlled. The actual circuit for the combination lock is shown in Fig. 10.

Operation of the lock begins with the reset mode. This is necessary because the reset can be initiated at any time in the event an incorrect digit is keyed. The output of a two-stage counter is decoded in the steering logic, and the BCD signals from the keypad are integrated into the counter's decoding logic so that a specific digit only can be passed through the enabling latches if both signals are coincident. It is mandatory that the four latches be set in the proper sequence (W,X,Y,Z) because any other combination will be defeated in the sequence detector.

A function table for the lock is given in Fig. 10. The 0 on the DEC IN line is the reset mode. The outputs of *FF1* and *FF2* assume a 0101 state. The *FF1* and *FF2* blocks are clocked flip-flops, with the clocking occurring on the trailing edge of the input pulse. The outputs of the keypad are fed to *IC4*, the outputs of which are selected to form the inputs to the associated NOR gates.

If the correct first digit is keyed in, line W goes to the high state, setting *IC5A/IC5B*. Both inputs to NOR gate *IC7A* are now low, setting the D input to *FF3* (*IC8A*) to high.

The sync pulse from the keypad has once more clocked the counter. If the second digit is correctly keyed in, line X goes high and sets the *IC5C/IC5D* latch. This clocks a low to one input of (*IC7B*). Once again, the keypad is operated with the correct digit to cause the associated latch to operate and placing a high on the Y line. This puts a low on the second input of *IC7B*. This sets the D input of *IC8B* to high.

The keypad is operated one more time with the final correct digit to set the Z line high. The Z latch clocks *IC8B* to change its output status. Either of the *IC8B* outputs can be used to interface to an external circuit.

If any of the four latches is set out of sequence, the clocking of *IC8A* and *IC8B* will be disrupted. The circuit is reset by operating the RESET switch.

Although the Fig. 10 circuit shows the use of a 1-to-10 decoder for the keypad input, a 1-of-16 decoder can be used for a hexidecimal input.

Switch Latch & Display. One difficulty with a keypad is that it is momentary. Once a key has been released, the action ceases. The addition of a quad latch, as shown in Fig. 11, will hold the switch outputs as long as dc power is applied. The *IC1* quad latch is used to drive BCD-to-7-segment decoder/driver *IC2* and a common-anode 7-segment LED display. This combination holds the last key depression and also produces a visible display of the digit depressed.

In Conclusion. In this article, we have described the major problems encountered when using mechanical switches—specifically keypad arrays—with digital circuits. We have offered some examples of how to deal with the problems and given hints on interfacing keypads with the electronic circuits. It is suggested that for further study and understanding of the material presented here you breadboard the circuits presented and do some experimenting on your own. ◇

Out of Tune

In "The Versatile Keypad" (August 1978) in Fig. 3, the left side of *IC11* was incorrectly labelled 3, 4, 5 from top to bottom. The correct sequence should be 3, 5, 4.

POPULAR ELECTRONICS

UPGRADE KEYPAD DESIGN

"The Versatile Keypad" (August 1978) was a fine article on keypads. I have a suggestion to add, however. One should consider using a 74C922 IC in the design. This chip decodes the switch matrix without diodes and features internal scanning and debouncing. It can also latch the most recent entry from the pad onto the output lines. The 74C922 is available from many mail-order sources for less than $6.00. A typical setup for this chip is shown in the diagram.—*Philip Thompson, Princeton, N.J.*