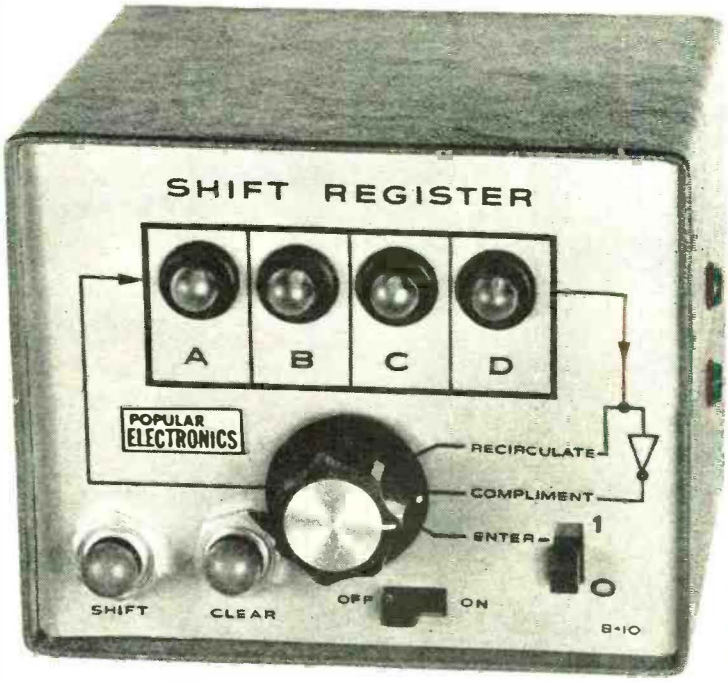


# Build a SHIFT REGISTER

*Storage for digital computers, computer logic*

BY DON LANCASTER

HAVE YOU EVER wondered how computers and electronic calculators perform arithmetic operations, or how they move data and numbers about? A unique circuit known as a "shift register" is responsible for these operations. The shift register is an electronic device that stores numbers, commands, words or locations when programmed to do so. Later, it "plays back" the stored information, either all at once or bit by bit until the register is empty or is back where it started from.

If you would like to experiment with the shift register, you can make one of your own at very little cost, using the instructions provided here. Your project can then be used as an entry in a science fair, a teaching aid, or simply as an interesting device for studying digital integrated circuits and computer logic.

This shift register employs three IC's and four transistors, arranged to form a "four-bit, serial-in/serial-out parallel-read" system. The functions provided are *enter*, *recirculate*, *compliment*, *shift*, and *clear*. The same project also demonstrates "walking ring counters" and "disallowed subroutines."

**Shift Registers in General.** Almost all digital computers and computer circuits

IC EXPERIMENTER'S CORNER

NO. 4 of 5

May, 1970

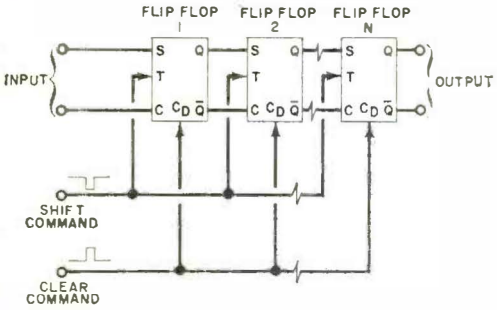


Fig. 1. Shift register is basically string of JK flip-flops connected in cascade. Any number of JK flip-flops can be connected in the manner shown.

are made up of simple elements that can have only two states: *on* and *off*, voltage and no voltage, or, most commonly, 1 and 0. The 1 or 0 represented at any time in a single element is called a *bit* (for binary digit). A string of related bits is a *word* (sometimes referred to as a *byte*). A word can represent a number (in binary, octal, binary coded decimal, decimal, or any other coding system), an address (a specific location or element in the machine), or an instruction (such as a "multiply" command).

A computer or calculator gets its words from a program on tape, cards, discs, drums, or a programmer (the person operating the computer). It then stores all the words it needs and later manipulates them as instructed.

The length of a word is simply the number of bits required to make up the word. The longer the word, the more accurate it can be. For example, for six-place accuracy using binary coded decimal (BCD) numbering, 25 bits are needed in each word.

The words are often stored in shift registers. A shift register comprises a number of stages, each of which can store one bit; 25 stages are needed to store a 25-bit word, and so on. There are several ways to get information into and out of a shift register, and there are several types of shift registers. For those of interest here, information is put in and taken out one bit at a time. This type is known as a "serial-in/serial-out" register.

The *shift* operation takes place when it is desired to move the stored word. On a shift command, every bit moves one—and only one—element to the right as the first element accepts a new bit from outside. (The sidebar on page 47 summarizes how this operation takes place.) The input can be an outside *enter* command, or the output *recirculated*, or the opposite of the output known as the *compliment* for special counter circuits.

Some shift registers, including this one, also have provision for a *clear* instruction that automatically puts all 0's in the register.

A shift register can be built with a train of conventional JK clocked flip-flops as shown in Fig. 1. Assuming that a *clear* command is first fed into the circuit, all stages are reset to a 0 condition. Upon receipt of a *shift* command, each stage passes its 0 one stage to the right, and the first stage accepts a new input.

Shift registers are then a means of accepting, storing, and later providing digital words when given the proper commands. In the *recirculate* mode, a shift register can march its word around bit by bit, produce one bit at a time as an output, and end up with the word right back where it started, ready for another use.

## PARTS LIST

*C1—0.1-μF disc capacitor*
*C2—100-μF, 6-volt electrolytic capacitor*
*I1-I4—NE-2 neon lamp*
*IC1—Dual two-input gate integrated circuit (Fairchild μL914)*
*IC2, IC3—Dual JK flip-flop integrated circuit (Motorola MC790P)*
*Q1-Q4—2N4400 transistor (or Motorola MPS6534)*
*R1-R6—470-ohm, ¼-watt resistor*
*S1—S.p.s.t. slide switch*
*S2—S.p.s.t. normally-open snap-action pushbutton switch*
*S3—Dual s.p.s.t. snap-action pushbutton switch, one pair of normally closed, other pair of normally open contacts*
*S4—Two-pole, three-position non-shorting rotary switch (Mallory No. 3223J)*
*S5—D.p.d.t. slide switch*
*Misc.—Printed circuit board (see text); dial-plate; 3¼" x 4" x 5" case; ¾" metal or fiber spacers (3); ⅜"-inner diameter rubber grommets for mounting lamps on front panel (4); #24 insulated wire for circuit board jumpers; control knob; #6 hardware; hookup wire; solder; etc.*

*Note—A metalphoto aluminum dial plate is available for $3 postpaid in U.S. from Reill's Photo Finishing, 4627 N. 11 St., Phoenix, AZ 85014. The following items are available from Southwest Technical Products Corp., 219 W. Rhapsody, San Antonio, TX 78216: Etched and drilled printed circuit board, #1726, $1.90; complete kit of parts including prepunched vinyl-clad case and dialplate, #172c, $9.75 postpaid in U.S.*

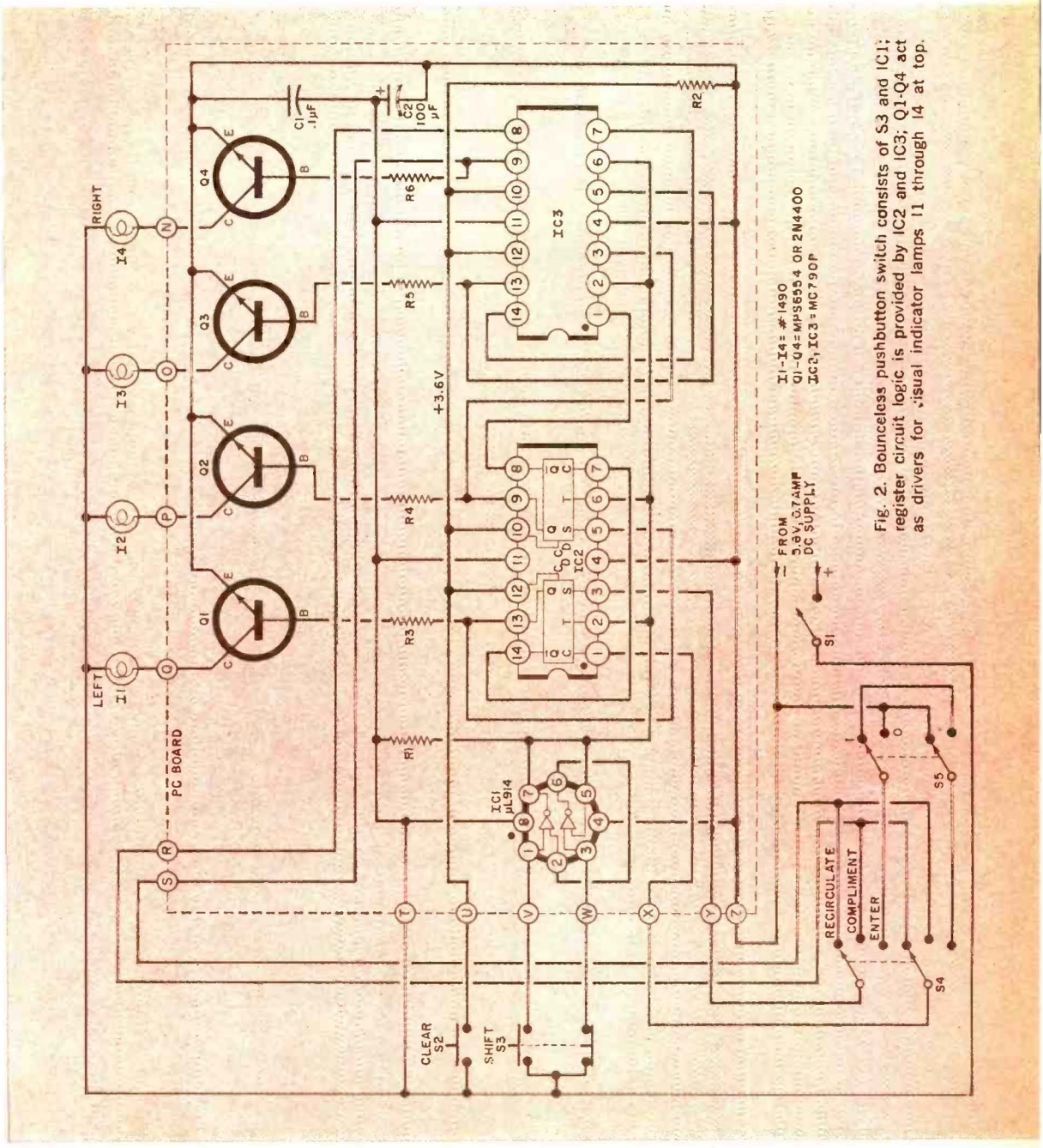


Fig. 2. Bounceless pushbutton switch consists of S3 and IC1; register circuit logic is provided by IC2 and IC3; Q1-Q4 act as drivers for visual indicator lamps I1 through I4 at top.

**How It Works.** Four JK flip-flops, *IC2* and *IC3* in Fig. 2, are used as the storage elements. These flip-flops are cascaded, with the first stage being driven from a source selected by switch *S4*. The source of the input data can be from switch *S5* (*enter*), the output (*recirculate*), or the opposite of the output (*compliment*).

A shift command is delivered to the toggle, or T, input of each stage from the bounceless pushbutton circuit made up of *IC1* and *S3*, which moves each bit one stage to the right for each shift command received. The clear operation is accom-

plished when *S2* delivers a positive voltage to the CD input of each stage, forcing the register into the 0000 condition.

The condition or state of each stage is indicated by lamps *I1-I4*. A lighted lamp indicates a "1" state, while an extinguished lamp indicates a "0" state.

**Construction.** A printed circuit board is a must for this project. The PC board can be purchased etched and drilled (see Parts List), or you can make your own by following the actual size etching guide provided in Fig. 3.

Component placement on the circuit

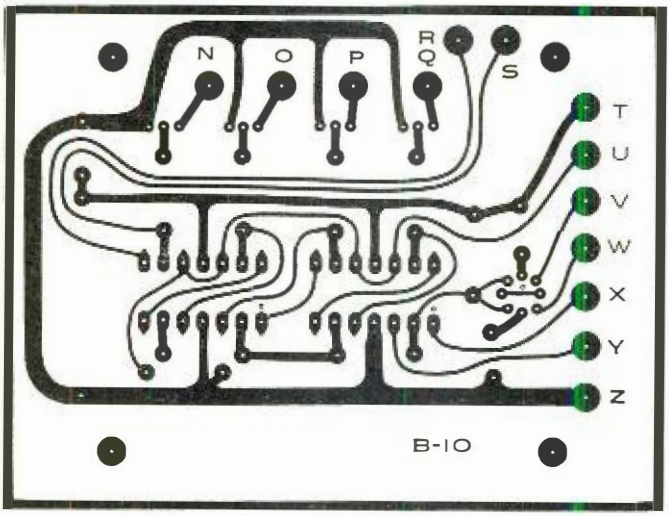


Fig. 3. In actual size printed circuit board etching guide, lettered contacts indicate off-the-board component connections.

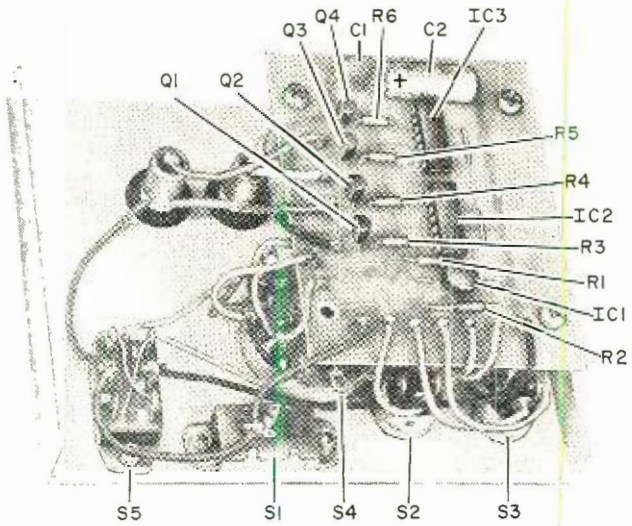


Fig. 4. When mounting integrated circuits on board, make sure indexing dot on IC1 is to left and notches of IC2 and IC3 are to bottom as shown in photo.

## HOW THE SHIFT REGISTER WORKS

**CLEAR:** With the clear command fed in, the register automatically resets to indicate 0000, regardless of the previous states. This operation empties the register, preparing it for its next use.

**ENTER:** The register places a selected 1 or 0 into the first stage on a shift command. All other 1's and 0's in the register then move one stage to the right. This is how a register is "filled" or "loaded."

Start with 0000
Enter a 1 1000
Enter a 1 1100
Enter a 0 0110
Enter a 0 0011

With each digit entry, a shift command must be initiated.

**RECIRCULATE:** The register shifts the word one stage to the right with each command received, with the last stage passing its 0 or 1 back to the first stage. This condition "marches" out a word bit by bit for outside use. When finished, the word ends up in its initial position if you shift exactly the number of stages in the register. This is how a word can be used but still retained.

Start with 0011
Shift to get 1001
Shift to get 1100
Shift to get 0110
Shift to get 0011

The binary word has gone "once around," one bit at a time appearing at the farthest right stage for outside use.

**COMPLIMENT:** This is a "trick" that is sometimes used to change a register into a counter. To compliment a register, the opposite of what is in the last stage gets passed back to the first stage, with all other stages passing their 0 or 1 one position to the right as usual. Eight shift commands are required to get the register back to where it started in a four-stage register.

Start with 0011
Shift for 0001
Shift for 0000
Shift for 1000
Shift for 1100
Shift for 1110
Shift for 1111
Shift for 0111
Shift for 0011

At this point, the register is back to where it started, taking eight shift commands to get it there. Hence, a divide-by-eight counter, called a "walking ring counter," is obtained.

board is shown in Fig. 4. When mounting components on the board, notice that *IC1* is identified by a flat and color dot near lead 8, while *IC2* and *IC3* are identified by a dot-and-code notch. Also take careful note of the lead orientations of transistors *Q1-Q4*. Use fine-grade solder and a small pencil soldering iron.

The lamps mount on the dialplate by press fitting them into ⅜" rubber grommets as shown in Fig. 4. The hookup wires are soldered directly to the contacts on the lamps, saving the price of individual lamp sockets. Next, the circuit board mounts on the front panel with the aid of three spacers. (Note: bend the lugs of *S4* so that they do not interfere with or touch the circuit board.

Once the components are mounted on the front panel and the circuit board is in place, interconnect with hookup wire, referring back to Fig. 2 as needed. Then mount the assembly in the vinyl-clad metal box provided with the kit of parts, or mount it in an aluminum utility box with dimensions at least 5" × 4" × 3½", and construction is complete.

When using the shift register, bear in mind that the circuit requires 3.6 volts at 700 mA, with less than 700 mV of ripple peak-to-peak. Any good bench supply will do. -30-