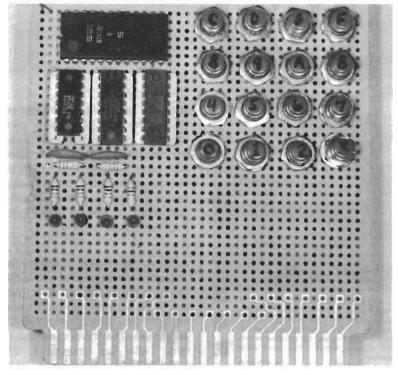


A HEXADECIMAL KEYBOARD ENCODER

The simple hex keyboard encoder described in this month's column can be significantly improved by adding a 4-bit register to store the hex code of the key that has been pressed. This means the LED readout will display the 4-bit code for a particular switch when the switch is initially closed and continue to display it after the switch has been released. The display will change only when the scanning circuit detects a new key closure.

Compare the complete circuit diagram for the hex encoder shown here with Fig. 7 and you'll note that a 74173 4-bit register has been added and the readout LEDs have been moved from the counter output to the register output.



The clock and scanner portions of the circuit have already been analyzed, and the 74173 was described in the March 1978 Experimenter's Corner. It's a flexible storage register with 3-state outputs that can be readily tied to the address or data

bus of a microcomputer or controller.

When the scanner circuitry detects a switch closure, the output of the 74150 sends a data enable pulse to the 74173 through one of the 7400 gates. The next clock pulse then loads the counter address into the 74173, and the circuit resumes its sequential scan of the switches. The four bits describing the previously closed switch, however, remain safely stored in the register.

The photograph shows a prototype version of the encoder assembled on a perforated board (Radio Shack 276-152 or equivalent). Note the extra space on the board for the addition of other circuits such as a RAM. Also note that a standard calculator keyboard was *not* used. These keyboards are inexpensive and readily available, but the switches are arranged in x-y format not compatible with this circuit. Instead, individual normally open pushbutton switches were used to make a custom hex keyboard.

I used wrapped-wire construction throughout with the exception of solder connections to the switch terminals, the LEDs and several of the resistor and capacitor leads. Total assembly time was about three hours. In a subsequent Project of the Month we'll add a 16-word RAM to the encoder. You'll find the resulting circuit very interesting, so be sure to consider building the basic encoder in the meantime.

