computer corner

Here's a look at the various ways you can interface a microprocessor to a 7 segment LED display.

J. TITUS, P. RONY, D. LARSEN and C. TITUS

REGARDLESS OF THEIR APPLICATION, MOST microcomputers need peripheral I/O devices for the input and output of data. The more common output devices include 7-segment displays, 5 × 7 dot-matrix displays, printers and CRT displays.

This month, we'll look at several methods that can be used to interface 7-segment displays, as well as different programs to "drive" such displays that are widely used in electronic games, calculators, point-of-sale (POS) terminals, gasoline pumps, children's toys and taxicab meters.

One of the simplest methods for interfacing a 7-segment display to a microcomputer consists of latching the appropriate data values from the data bus under software control (see Fig. 1). The latch's inputs (7475) are wired to the computer's bidirectional data bus, and the latch's outputs are wired to a 7-segment decoder/driver (7447). The decoder's outputs (current-sinking) are wired to the 7-segment display with 220-ohm current-limiting resistors. When an

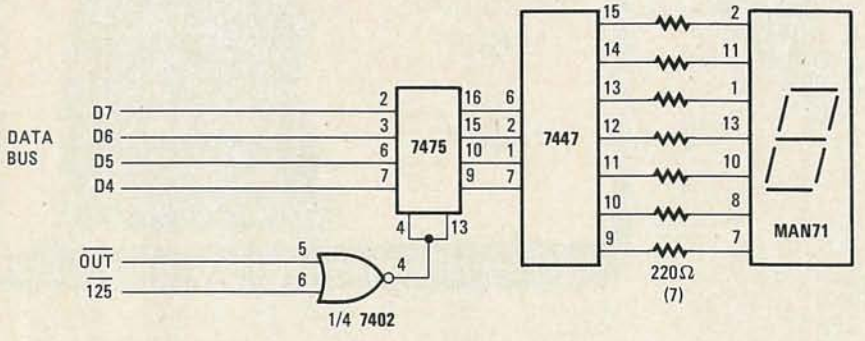

Fig. 1

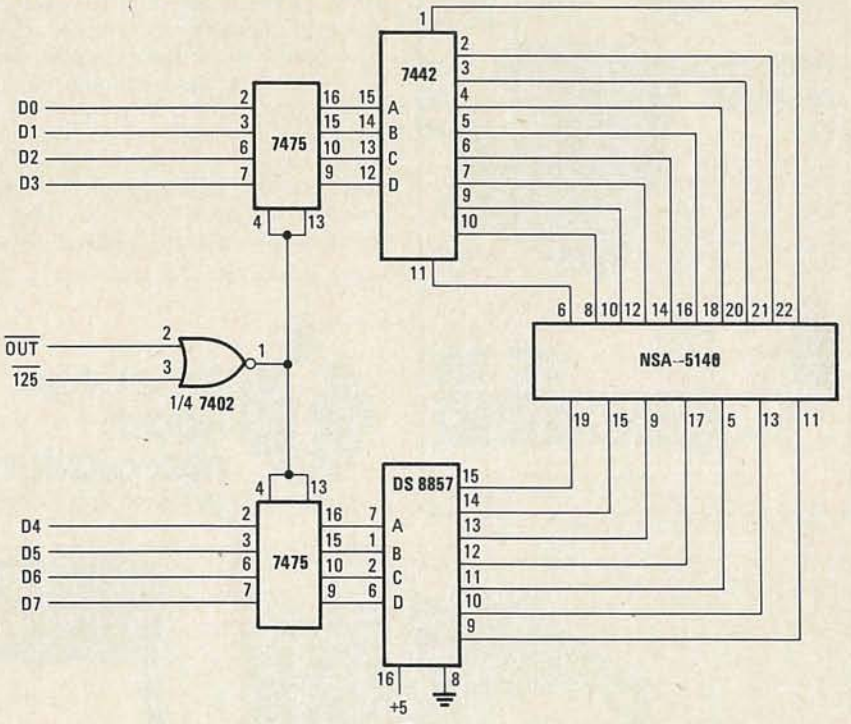

Fig. 2

OUT125 instruction is executed, the content of the 8080's A register is latched by the display interface, and the two BCD numbers represented by D7–D4 and D3–D0 are illuminated on the two displays. The instructions listed in Table 1 cause a "39" to be displayed.

To display a 10-digit number using this method, 10 latches, 10 decoder/drivers, 70 resistors and 10 seven-segment displays will be required. One method of reducing the "parts count" for this interface is to use a device such as the 8255 Programmable Peripheral Interface (PPI) integrated circuit. This device can be used as three independent 8-bit output ports, so it is the equivalent of six 7475 latches. Therefore, two 8255 PPI IC's, along with 10 decoder/drivers, 70 resistors and 10 seven-segment displays would be required in the interface. An obvious disadvantage of this method is the large number of IC's required. However, one advantage is that the software needed to drive this interface is relatively simple. Also, the microcomputer only has to output this information once to the interface for the information to be continuously

TABLE 1—DISPLAYING A 39 on the two-digit display

```
/THIS SECTION OF A PROGRAM OUTPUTS THE
/BIT PATTERN 00111001 (OCTAL 071, HEX 39)
/TO AN OUTPUT PORT EQUIPPED WITH TWO
/SEVEN-SEGMENT DISPLAYS.
    .
    .
MVIA    /LOAD A WITH THE FOLLOWING IMMEDIATE
071     /DATA BYTE (HEX 39, BINARY 00111001)
OUT     /OUTPUT IT TO THE TWO SEVEN-SEGMENT
125     /DISPLAYS EQUIPPED WITH LATCHES
    .   /CONTINUE WITH THE REMAINDER OF THE
    .   /PROGRAM
```

displayed. This, of course, is due to the latches or the 8255 IC's in the interface. Thus, the microcomputer can output numeric information once and then go on to perform any other required operations.

Another interfacing method is digit *multiplexing*. Multiplexing reduces the display-interface electronics (number of parts) to a minimum; however, at the expense of longer and more complex display-driver software. Multiplexing a display consists of enabling or turning on one particular digit with a *digit-enable code* and providing the BCD numeric information for that digit to a multidigit display interface. In this way each digit is turned on, one at a time, as the actual BCD data for each digit is provided. Multiplexing is usually only used with multidigit displays.

As an example, let's suppose the number "237" is to be displayed on a 3-digit multiplexed display. To display this number the BCD value for the digit "7" would be output to the interface along with the digit-enable code for the right-hand display. After a short period (1 μs to 10 μs) the BCD value for number "3" would be output along with the digit-enable code for the middle digit. Again, after a short delay, the BCD value for the number "2" and the digit-enable code for the left-hand display would be output to the interface. By performing this sequence 50 or more times every second, each digit in the display *appears* to be on *all the time*. This display method is used in hand-held calculators. Even though the digits are being turned on and off, it happens too fast for the eye to see. The interface for a 10-digit multiplexed display is shown in Fig. 2.

When an OUT125 instruction is executed, bits D3–D0 of register A determines which one of the 10 digits in the display will be enabled (turned on). Therefore, these 4 bits constitute the digit-enable code. Bits D3–D0 are latched (7475) and are decoded with a one-of-10 decoder (7442). The decoded outputs of the 7442 are wired to the common cathodes of the individual digits in the display. Bits D7–D4 provide the BCD code of the value to be displayed (0–9). These bits are also latched (7475) and are decoded by a 7-segment decoder/driver (DS8857, manufactured by National Semiconductor, Santa Clara, CA):

An additional display method involves the use of an external display controller IC to control the multiplexed display. The Intel Corporation manufactures some of these IC's that are compatible with the 4004/4040, 8080 and 8085; these IC's are the 4269, 8279 and 8279-5. National Semiconductor Corporation also has developed two display controller IC's that can be used with 6-digit displays. One of the devices (the MM74C912) can be used to display 0–9 and the other (the MM74C917) can be used to display hexadecimal numbers. **R-E**