

# Simplifying sum-correction logic for adding two BCD numbers

by Robert D. Guyton  
Mississippi State University, Mississippi State, Miss.

To add two numbers in binary-coded decimal form, much less logic hardware is needed if one of the numbers is converted to the excess-6 binary code before the addition is done. The other number remains unchanged.

The block diagram of (a) outlines the approach. One BCD input is converted to the excess-6 code by a second-level logic circuit, which is drawn in (b). This translated number and the unchanged BCD number are then added by a 4-bit binary adder. The resulting output

carry is correct, but the sum must still be corrected—and can be corrected by a simple second-level logic circuit, rather than a multilevel type of logic circuit based on half and full adders.

The BCD-to-excess-6 translator circuit needs four NAND gates and three inverter gates. The rest of the over-all addition circuit is shown in (c): the four-bit binary adder requires four full adders, while the sum-correction circuitry requires 10 NAND gates and three inverter gates.

The complete excess-6 addition circuit, therefore, consists of 14 NAND gates, six inverter gates, and four full adders. As against an addition circuit based on excess-3 code conversion, that's a savings of six NAND gates, three inverter gates, one full adder, and two half adders. □

Designer's casebook is a regular feature in Electronics. We invite readers to submit original and unpublished circuit ideas and solutions to design problems. Explain briefly but thoroughly the circuit's operating principle and purpose. We'll pay \$50 for each item published.

**Conserving logic hardware.** The circuit for adding two binary-coded-decimal numbers can be implemented with fewer devices by changing one of the BCD numbers to the excess-6 code format. When this conversion is done, simple logic gates can be used to perform the necessary sum correction. The figure shows the circuit's block diagram (a), the excess-6 code translator (b), and the complete circuit (c).

