Relays Do Simple Arithmetic

Part III—How to use relay adding circuits for subtraction and multiplication in the binary system

By EDMUND C. BERKELEY* and ROBERT A. JENSEN

IN THE two earlier articles we have seen how an electric brain can:

1. store information in a register;
2. transfer information from one register to another; and
3. add two numbers expressed in binary notation (the scale of two).

Being interested in constructing a relay calculator, in this article we shall consider subtraction and multiplication using relays.

We shall keep to binary numbers for the present for three reasons: It is easy to carry out the operations we are interested in. Also, binary notation is good for electron-tube calculating circuits as well as relay calculating circuits. Finally, it is a good introduction to the circuits needed for calculating in the decimal scale.

Suppose we wish to subtract the binary number 101 (read "one-oh-one," meaning one 4 plus no 2's plus 1, or 5) from the binary number 1110 (read "one-one-one-oh," and meaning one 8 plus one 4 plus one 2 plus no 1's, or 14). We write 101 under 1110 and subtract:

$$\begin{array}{r} 1110 \\ 101 \\ \hline 1001 \end{array}$$

How do we manage to subtract? We recall the binary addition table:

	0	1
0	0	1
1	1	10

Then we say under our breath, for the first column at the right: "1 from 0 does not go, borrow 1; 1 from 10 (read "one-oh" not "ten") is 1, write down 1." For the next column, we say, remembering the borrow: "0 from 0, write down 0." For the third column: "1 from 1 is 0, write down 0." For the last column: "nothing from 1 is 1, write down 1." The result is 1001 (read "one-oh-oh-one," meaning one 8 plus no 4's plus no 2's plus one 1, or 9) just as we would expect it to be.

We could set to work and design a circuit which would reproduce this process and give the precise result we desire. But isn't there an easier way?

There is an easier way to subtract—by using the addition circuit shown in the last article, and using the mathematical fact that subtracting a number is the same as adding the complement.

To make the idea of complement clear, let us return for a moment to decimal notation (the scale of 10) and consider a desk adding machine having just five columns. Suppose we consider a number 864 (eight 100's plus six 10's plus four 1's). Suppose we set the machine at 0 and subtract 864. We will obtain 99136. This is called the *complement* of 864 (also called the *tens complement* of 864). For, if we take 864 and 99136, and add them, we get 100,000; but the extreme left-hand digit (the 1) being beyond the capacity of the five-column adding machine, it vanishes and the result is 00,000 or zero. (In a machine of ten columns instead of five the complement would be 9,999,999,136, correspondingly.)

Now to subtract 864 from any number—suppose it is 3,145—we add the complement:

$$\begin{array}{r} 3145 \\ -\ 864 \\ \hline 2281 \end{array} \qquad \begin{array}{r} 3145 \\ +\ 99136 \\ \hline 102281 \end{array}$$

The extreme left-hand digit in 102,281 will disappear off the machine, giving 2,281 as the result, which is correct.

The complement (such as 99,136) is easily found for any number (such as 00,864) by two rules:

1. take each digit away from 9 (obtaining what is called the *nines complement*, in this case 99,135);
2. add 1 to the result (obtaining in this case 99,136, the *tens complement*).

What is the analogue in binary notation to these complements in decimal notation? In decimal notation we have a *nines complement*, nine being one less than ten, the base of the scale of notation; so, in binary notation, we shall have a *ones complement*, since one is one less than two, two being the base

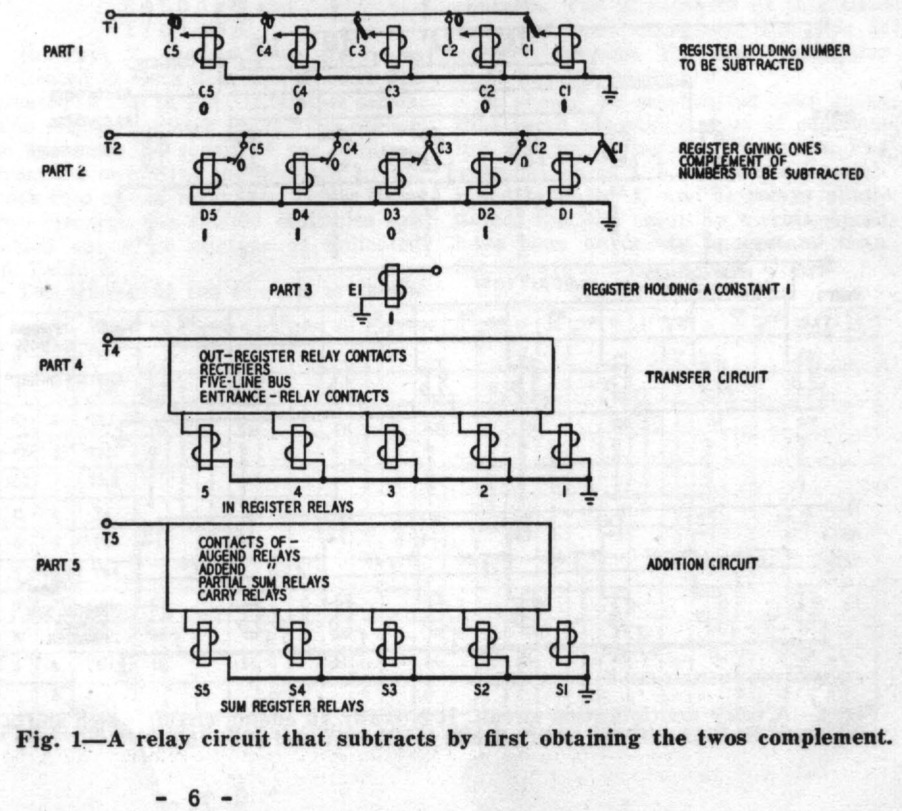

Fig. 1—A relay circuit that subtracts by first obtaining the twos complement.