How an Electric Brain Works

Part IV—Long division with relays—our little electric brain learns how to divide and to convert decimal numbers to binary and back again. Simon is getting an education

By EDMUND C. BERKELEY* and ROBERT A. JENSEN

PREVIOUS articles of this series have shown how an electric brain made of relays can add, subtract, and multiply.

Now we shall carry out division. As before, we shall consider the process in *binary notation*, the scale of two.

As a second topic, we shall consider how to make a relay calculator convert a number from decimal notation to binary notation, and back again. There is every reason in the world why the machine itself should convert any decimal number, say 23, into the corresponding binary number (in this case 10111, one-oh-one-one-one, or one 16 plus no 8's plus one 4 plus one 2 plus one 1).

Addition, subtraction, and multiplication turned out to be very simple in binary notation as compared with decimal. The same is true with division: binary division is simple as can be.

Suppose we divide 1101 (one-one-oh-one, or 8 plus 4 plus 1, or 13 in decimal) into 10000101 (one-oh-oh-oh-oh-one-oh-one, or 128 plus 4 plus 1, or 133.

We do this in the same general way as we do in decimal division, except that we act as if we knew only the two digits 1 and 0:

```
                  01010    (Quotient)
(Divisor)  1101)10000101   (Dividend)
                0000
                10000 (1st Partial
                 1101   Remainder)
                 0111 (2nd Partial
                 0000   Remainder)
                  1110 (3rd Partial
                  1101   Remainder)
                  0011 (4th Partial
                  0000   Remainder)
                   011 (Remainder)
```

Only two multiples of the divisor are used, one times the divisor, and zero times the divisor—and the latter is of course zero in every digit. No other multiples of the divisor are needed. If we simply compare the divisor with the partial remainder at any point in the division, we can tell whether the digit of the quotient is 1 or 0.

Circuits for division

As before, to keep the circuits simple, let us ignore a number of fine points, such as: fractions; the *binal* point (the analogue in the scale of two of the decimal point in the scale of ten); positive and negative numbers; size of numbers; etc. Suppose that we have an eight binary digit dividend, and a four binary digit divisor.

The circuit is on the opposite page. In part 1, terminal T1 is energized at the start, and holds up the relays storing the dividend through their hold contacts. (All current-carrying circuits and relay contacts in the energized state are in red.) The actual number which these relays store, of course, depends on something that happened before the time at which we begin. In the same way, the divisor is stored in relays of part 2 of the circuit, and terminal T2 holds them up.

Now different things have to happen at different stages during the division. So we want to have some relays that will tell us at what stage we are during the process of the division. This is the function of the K relays of part 3 of the circuit. The stages that they detect and report are 0,1,2,3,4. The time chart in Fig. 1 shows that stage 0 lasts from time 1 to time 8, stage 1 from times 9 to 16, stage 2 from times 17 to 24, etc. At stage 0, we attend to the first quotient digit; at stage 1, we attend to the second quotient digit; etc. The red parts of the circuit apply to the first stage of the division only.

We have to start off the divisions by selecting some digits, which we can call a *partial remainder* (see part 4). At stage 0, this is the first four digits of the dividend; but at later stages this is the result of a subtraction together with "bringing down" one more digit of the dividend. The circuit of part 4 shows that at each stage of the division, we have just the partial remainder that we desire stored in the E relays. We have to look ahead to part 8, of course, and take on faith that the G relay contacts in part 4 will express the result of a subtraction that we want.

The next thing that we must do is decide whether the divisor "goes" into the partial remainder, or whether it "doesn't go". To make this decision, we must compare two numbers and decide which is the larger. The divisor "goes" and yields 1 as a digit of the quotient if, and only if, the partial remainder is larger. A circuit that does exactly this is shown in part 5. The red contacts show the original partial remainder (stored in the E relays) and the divisor (A relays). We see that there is no path for the quotient relay Q to be energized, and so the first "quotient digit" is 0.

Before we go any further, we want to store that quotient digit, so that we shall know the whole quotient when we get through with the division. This duty is performed by the circuit of part 6, which shows how the digit quotient is routed, according to the time it is obtained, into the right C relay.

We now want to determine the multiple of the divisor that depends on the quotient digit and the divisor. This is the function of part 7 of the circuit, which will give us the divisor itself if the quotient digit is one and zero in all digits if the quotient digit is zero.

In part 8 of the dividing circuit, the subtraction of the divisor multiple from the partial remainder is indicated schematically, because actual circuits for subtraction were discussed previously.

The timing of the circuits, up to the end of the first two quotient digits, is shown in the timing chart of Fig. 1. The same conventions are used here as in the time chart for multiplication in the previous article. Successive time

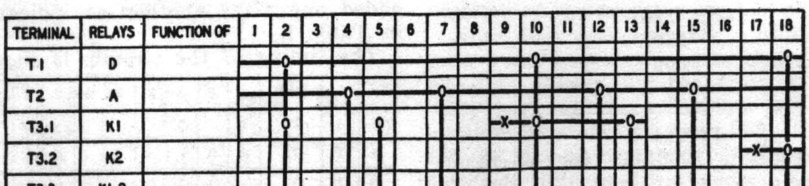

Fig. 1—Timing chart which shows the sequence of operation for the first two stages of the division with binary numbers performed by the circuit of Fig. 1.