bered terminal 8. The result of the multiplication, the product, comes out on the output line, and the two factors of the multiplication, the multiplicand and the multiplier, come in on lines T1 and T2.

All information travels through this circuit in pulses serially along one line, and so the scheme of the circuit could operate for binary numbers as large as 50 or 100 digits, just by altering the length of the delay lines. To explain the circuit operation, we shall assume that we are going to multiply two four-digit binary numbers. We shall multiply binary number 1101 (one-one-oh-one, or one eight plus one four plus no two's plus one one, or thirteen) by the binary number 1011 (one-oh-one-one, or one eight plus no four's plus one two plus one one, or eleven). The answer will be 10001111 (the sum of 128, 8, 4, 2, and 1, or 143, which of course equals 13 times 11). Since the answer will be eight binary digits long it is convenient to specify that the multiplier circuit operates in six machine cycles, each handling eight pulses corresponding to eight binary digits. The reason for eight pulses is obvious; that for six cycles will appear later.

This example of multiplication, by the way, is the same example worked out with relays that was discussed in the December, 1950, article of this series, Part III. The comparison of the multiplication there using relays and the multiplication here using electronic tubes is instructive, if any reader cares to look it up.

In the first cycle, the multiplicand 1101, in the form of a series of eight pulses, or a binary number, 00001101, enters along line T1. The number shows the timing of these pulses *represented in order from right to left, the earliest pulse being at the right.*

Note well that on any line where the direction arrow points to the right, a regular binary number will represent pulses in the proper time sequence. We have avoided writing binary numbers, or series of pulses, on lines directed to the left because the sequence of pulses and the regular way of writing digits in the number would be opposite. For later cycles, we write the binary numbers of eight digits one under another. This helps to see without difficulty the pattern of as many as 40 or 48 consecutive pulses.

The series of eight multiplicand pulses go into the 8-pulse-time delay line D1, and they will accordingly issue in cycle 2 as 00001101, and apply for admission at the door of AND circuit A1, EXCEPT circuit E1, and EXCEPT circuit E2.

FLIP-FLOP F-F1 controls admission to the EXCEPT circuits E1 and E2. Because of the pattern of reset pulses (see later discussion) coming to this flip-flop, the EXCEPT circuit E1 is inhibited, while E2 is not inhibited. Consequently, the pulses go through E2, and through the one pulse-time delay line D3, through the noninhibited EXCEPT circuit E3, and round the loop through D1 again. As a result, *the multiplicand is shifted, to the left one digit* and the output from the delay line D1 of the numbers applying for admission to the AND circuit A1 is therefore as follows:

Cycle	Number
1	00000000
2	00001101
3	00011010
4	00110100
5	01101000
6	00000000

In the first cycle also, the *multiplier* 1011 enters as a series of pulses 0000-1011 into the one pulse-time delay line D4 and the seven pulse-time delay line D2, and goes round the loop through delay line D2 in such a way that there issues from the loop the following numbers:

Cycle	Number
1	00000000
2	10001011
3	01100010
4	11000101
5	10110001
6	00000000

The first digit is carried around to the eighth digit's place by D2 and E4 in each cycle. Because the delay line is only 7 pulses long it advances the pulses one step. In cycle 3, for instance, we no longer have the number 1011 at the right-hand side, but 101, the other one having been attached to the end (left side) of cycle 2, where it replaces the zero that would otherwise be there. The *first* digit in any one of these numbers, in the sequence in which they pass through the circuits, is at the *right*, and the *last* one at the *left*. This follows our regular practice in adding or multiplying on paper, but not in writing down figures. These numbers now continue and apply for admission at the door of AND circuit A2.

The other line going into AND circuit A2 is line T3, called "timing-pulse input from time-pulse selector," and the pulses (or numbers) that come in on this line are as follows:

Cycle	Number
1	00000000
2	00000001
3	00000001
4	00000001
5	00000001
6	00000000

These are standard pulses and would accordingly apply for any four-diget multiplier.

Their purpose is, we shall see later, to allow a single digit of the multiplier to go through the AND circuit, and the timing is such that just the proper digit of the multiplier is allowed to go through the AND circuit.

Now an AND circuit performs the operation of "both," i.e., logical multiplication, operating digit by digit, pulse by pulse. Consequently, the numbers that issue from the AND circuit are the logical product, according to the following very simple table, of the numbers coming in:

	0	1
0	0	0
1	0	1

So the numbers (or pulse series) coming out of AND circuit A2 are now:

Cycle	Number
1	00000000
2	00000001
3	00000001
4	00000000
5	00000001
6	00000000

The output of AND circuit A2 leads to the "set" side or left side of FLIP-FLOP 2. The "reset" side or right side of FLIP-FLOP 2 receives a series of reset pulses (see later discussion) as follows:

Cycle	Number
1	10000000
2	10000000
3	10000000
4	10000000
5	10000000
6	10000000

These are the two inputs of the flip-flop. So what comes out?

A flip-flop conducts on one side and not on the other side, depending on the last pulse that has come in on either side. Specifically:

Input: Last Pulse Received on		*Output:* Conducting on	
Left Side	Right Side	Left Side	Right Side
1	0	1	0
0	1	0	1

The operation of a flip-flop is ambiguous, not defined, if the last pulse comes in simultaneously on both left and right sides. So before any inputs are acceptable to a flip-flop there must be no cases of 1 and 1 occuring at the same time.

Applying this rule we can calculate the output of FLIP-FLOP 2 as follows:

	Input		*Output*	
Cycle	Left Side	Right Side	Left Side	Right Side
1	00000000	10000000	0???????	1???????
2	00000001	10000000	01111111	10000000
3	00000000	10000000	01111111	10000000
4	00000001	10000000	00000000	11111111
5	00000001	10000000	01111111	10000000
6	00000000	10000000	00000000	11111111

Wherever we see a zero on one side of the flip-flop output, we must see a one on the other side. This is merely equivalent to saying that one side or the other, but not both sides, of the circuit are conducting at any given time.

We may note that the output of the right side of FLIP-FLOP 2 is not used in this circuit, but is available and might be useful somewhere else in the computer. The output of the left side, however, goes to AND circuit A1 and is there very useful. The flock of 1's on cycles 2, 3, and 5 allows the shifted multiplicand to go through A1 in the case of the first, second, and fourth multiple, and this is precisely what we want. Furthermore, the fourth multiple of the multiplicand will never be more than 7 digits, and so the change of the flip-flop at time 8 in each cycle does not interfere with the multiples of the multiplicand.

ADDER at cycle 2 takes in the first multiple of the multiplicand, adds it to 0 coming in on the other line, and passes it around the loop through the noninhibitory EXCEPT circuit E5 to the partial sum register, the 8 pulse-time delay line D5. One cycle later this first multiple comes out, going to the adder. But now it is matched in time (cycle 3) with the second multiple, and the adder produces the sum of the two of them. and