CONTROL and surrounded by a dashed line is the group of circuits which control the gates all through the machine, both outside itself and inside. The unit marked OPERATION FUNCTION TABLES converts various pieces of information, particularly instructions that program the operation of the computer, into gate controls. An example of this kind of situation in a relay computer is shown in article V and in an electronic computer is shown in article XI. Control over the connection of circuits to the bus results in the control of the flow of information through the machine.

Programming

The term *programming* is now generally used to mean giving or arranging a sequence of instructions under which an automatic sequence-controlled calculator operates. In Simon, and other "old-fashioned" automatic computers, the sequence of instructions is external to the machine, comes along on a loop of tape, and is read order by order. But in many of the new computers, and in modified examples of the old computers (including the Harvard IBM Mark I machine at Harvard University), the instructions are almost all internal, within the machine, and can be called for by the machine itself as it requires them.

Typically, the control of one of these newer types of machines is in a register called the "program register" or the "control register." On each machine cycle this register contains an "order," information which is interpreted by the machine to control the opening and closing of the gates (such as shown in Fig. 5). In general an order consists of three parts:

n, the number of the order, which will generally be from 000 to 999.

k, the kind of order, such as "transfer into . . .," and which may be denoted by 0 to 9.

r, the number of a register referred to, which may be from 000 to 999, if the machine has a thousand registers.

Many programs can be covered in 400 orders, and they may be stored in 400 of the 1,000 registers. The other 600 registers are used to store numbers involved in the calculation. The machine is designed so that regularly order number n is stored in register number n.

Necessary orders

Now what are the kinds of orders sufficient to carry on the functioning of a modern versatile electronic computer? They are surprisingly few. See the order chart.

The list shown in Chart 1 may be made longer or shorter. We have here used the device of calling for operation specifications out of certain storage registers. In this way the machine may be made very versatile. On the other hand, by lengthening the list of kinds of orders, to include directly addition, subtraction, multiplication, and other operations among the kinds of orders, some machine time can be saved, and some programming directness gained.

CHART I

If k equals:	*Machine Performs Operation:*	*and For the Next Order Goes to:*
0	Transfer operation in register r to computer's *operation* register	Next numbered order
1	Transfers number in register r to computer's register A	"
2	Transfers number in register r to computer's register B	"
3	Transfers number in computer's result register C to register r	"
4	Transfers number on input tape to register r	"
5	Transfers number in register r to output tape	"
6	None	The order numbered r
7	None	The order numbered r, provided some special register (No. 069 for example) contains 1; otherwise the next number numerically.
8	None	The order number stored in register r, provided special register 069 contains 1; otherwise, the next order numerically.
9	Stops	

The most important of all the orders is the conditional transfer of program, kind of order 8 (also to some extent, kind of order 7). This order is the heart of flexible automatic control, and is one of the main reasons for asserting that these machines are intelligent—are brains—because they can adjust their activity to indications revealed as they compute.

In the next and last article of this series, we shall discuss the application of a good deal of what we have been talking about, to the Bureau of Standards Eastern Automatic Computer, SEAC.

—end—