

THE EDSAC

The EDSAC (Electronic Delay Storage Automatic Calculator)

1. Introduction. The EDSAC is a serial electronic calculating machine now in operation in the University Mathematical Laboratory, Cambridge, England. It works in the scale of two and uses ultrasonic tanks for storage. The main store consists of 32 tanks each of which is about 5 ft. long and holds 32 numbers of 17 binary digits, one being a sign digit. This gives 1,024 storage locations in all. It is possible to combine two adjacent storage locations so as to accommodate a number with 35 binary digits (including a sign digit); thus at any time the store may contain a mixture of *long* and *short* numbers. Short tanks which can hold one number only are used for the accumulator and multiplier registers in the arithmetical unit, and for control purposes in various parts of the machine.

A single address code is used in the EDSAC, orders being of the same length as short numbers. Most orders consist of a functional part which defines the operation and a numerical part which defines a storage location; some orders, however, consist of a functional part only. The complete order code is as follows.

EDSAC Order Code

Order	Explanation
<i>A n</i>	Add the number in storage location <i>n</i> into the accumulator.
<i>S n</i>	Subtract the number in storage location <i>n</i> from the accumulator.
<i>H n</i>	Transfer the number in storage location <i>n</i> into the multiplier register.
<i>V n</i>	Multiply the number in storage location <i>n</i> by the number in the multiplier register and add into the accumulator.
<i>N n</i>	Multiply the number in storage location <i>n</i> by the number in the multiplier register and subtract from the contents of the accumulator.
<i>T n</i>	Transfer the contents of the accumulator to storage location <i>n</i> , and clear the accumulator.
<i>U n</i>	Transfer the contents of the accumulator to storage location <i>n</i> , and do not clear the accumulator.
<i>C n</i>	Collate the number in storage location <i>n</i> with the number in the multiplier register, i.e., add a 1 into the accumulator in digital positions where both numbers have a 1 and a 0 in other digital positions.
<i>R 2ⁿ⁻²</i>	Shift the number in the accumulator <i>n</i> places to the right, i.e., multiply it by 2^{-n} .
<i>L 2ⁿ⁻²</i>	Shift the number in the accumulator <i>n</i> places to the left, i.e., multiply it by 2^n .
<i>E n</i>	If the number in the accumulator is greater than or equal to zero, execute next the order which stands in storage location <i>n</i> ; otherwise, proceed serially.
<i>G n</i>	If the number in the accumulator is less than zero, execute next the order which stands in storage location <i>n</i> ; otherwise, proceed serially.
<i>I n</i>	Read the next row of holes on the tape, and place the resulting 5 digits in the least significant places of storage location <i>n</i> .
<i>O n</i>	Print the character now set up on the teleprinter, and set up on the teleprinter the character represented by the five most significant digits in storage location <i>n</i> .
<i>F n</i>	Place the five digits which represent the character next to be printed by the teleprinter in the five most significant places of storage location <i>n</i> .
<i>Y</i>	Round off the number in the accumulator to 34 binary digits.
<i>Z</i>	Stop the machine, and ring the warning bell.

Ordinary 5-hole telegraphic punched tape is used for input. Each row of holes represents a 5-digit binary number and the basic input operation is to transfer this number to the store. Similarly, the output mechanism is a teleprinter, and the basic output operation is to transfer a 5-digit binary number to the printer, and to print the corresponding character. The teleprinter code is chosen so that binary numbers up to nine are printed as the corresponding figures and a similar code is used for input. This enables the operation of conversion to and from the decimal system to be programmed as part of the calculation.

The purpose of the *F* order is to enable the operation of the printer to be checked. Apart from this, no special checking facilities are provided in the EDSAC, and it is left to the programmer to incorporate in the program such checks as he considers necessary.

2. The Control Sequence. When the machine is in operation, orders are executed automatically in the order in which they stand in the store. However, when a conditional order (*E* or *G*) is encountered and the condition is satisfied, the next order to be executed is the one which stands in the storage location specified in the conditional order.

Count is kept of the orders as they are executed by means of a short tank—known as the sequence control tank—which has associated with it an adding circuit through which unity is added to the number stored in the tank each time an order is executed. During a conditional order (when the condition is satisfied) unity is not added, but, instead, the number in the tank is replaced by the numerical part of the conditional order.

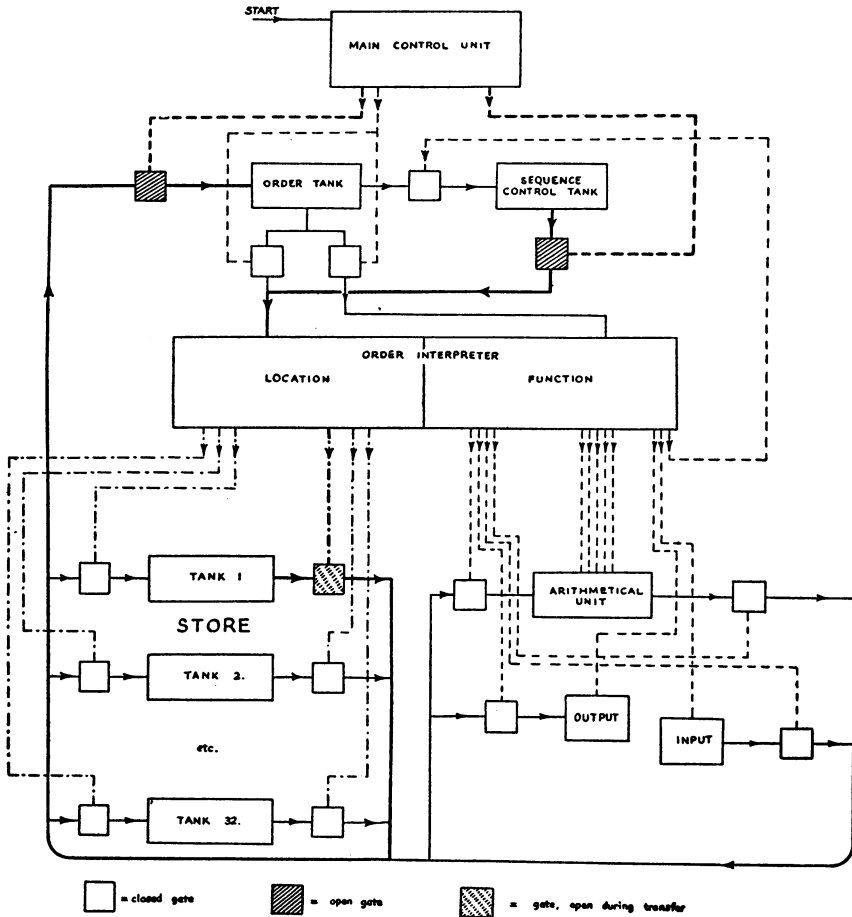
The control sequence of the machine falls into two parts. In stage I, an order is transferred from the location in the store given by the number in the sequence control tank to a short tank known as the order tank. In stage II the order in the order tank is executed.

The various constituent units of the machine—store, arithmetic unit, input unit, output unit, sequence control tank, and order tank—are connected together through gates, so that the interconnections proper to each successive part of the control sequence can be made by opening the appropriate gates. The gates are operated by waveforms supplied either by the main control unit or by a part of the machine called, for the purpose of this description, “order interpreter.” In the machine itself the “order interpreter” consists of a number of separate units. In the diagrams that follow, wires which carry pulses are shown as continuous lines, while those which carry control waveforms are shown as dotted lines.

Figure 1 shows the state of the machine during stage I of the control sequence. Here it is assumed that the storage location specified by the number in the sequence control tank is in tank 1 of the main store. The wire along which the order passes from this storage location to the order tank is shown by a heavy line in the diagram, and similarly the wires along which the control waveforms pass are shown as heavy dotted lines. The location section of the “order interpreter” receives the number in the sequence control tank and emits a waveform which opens the output gate of Tank 1. There are 32 orders (or short numbers, or a mixture of orders, short numbers, and long numbers) circulating in the tank. The gate is opened as soon as the required number becomes available and closed again when it has passed through. In this respect it is treated differently from the other gates in the machine

which remain open or closed during the whole of stage I; The difference is indicated in the diagram by the different form of shading used. The switching and timing necessary for the operations detailed above are carried out within the "order interpreter."

Stage II is very similar to stage I except that the type of operation performed is not restricted to a simple transfer from the store to the order tank. During stage II, the order tank is connected to the "order interpreter"



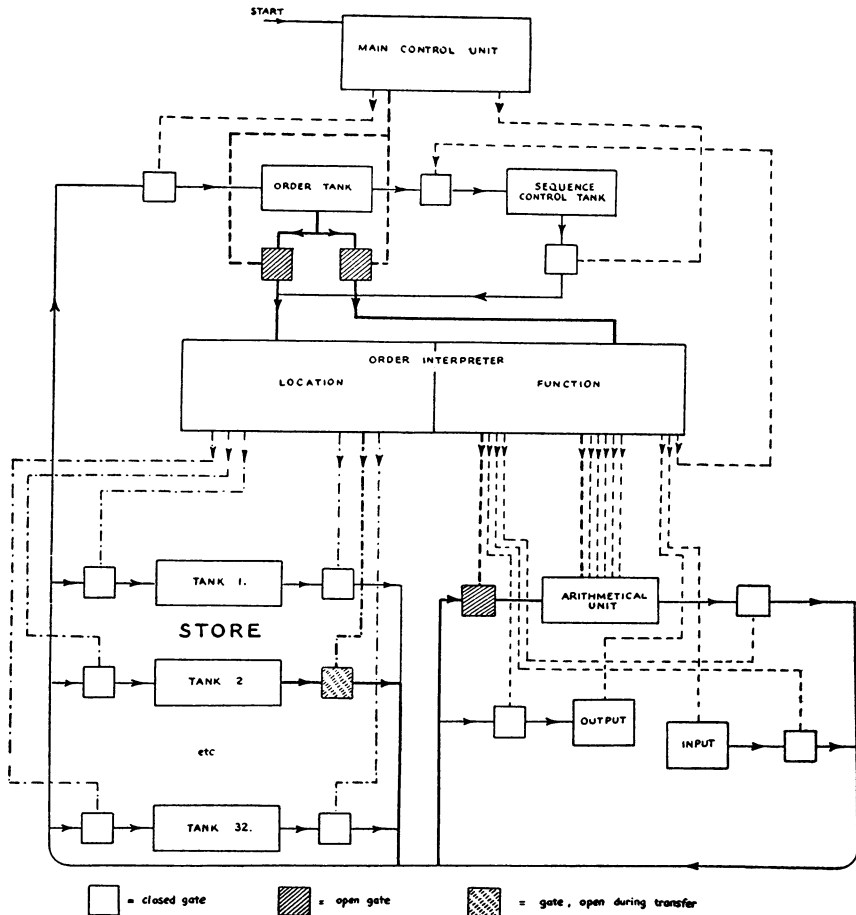
STAGE I.

FIG. 1

in place of the sequence control tank, and the numerical part of the order is dealt with in the same way as the number in the sequence control tank during stage I. A new feature, however, is that the function section of the "order interpreter" comes into action and interprets the functional part of the order; it proceeds to emit waveforms which (a) set up connections between the units ready for the passage of numbers and (b) place the arithmetical unit (or input or output units, as the case may be) in a state of readiness for executing the order. Figure 2 shows the various gates set up for the

execution of an add order calling for the transfer of a number from tank 2 of the main store to the arithmetical unit. One of the control wires passing from the "order interpreter" to the arithmetical unit is shown as a heavy (dotted) line to indicate that the arithmetical unit has been prepared for an addition.

The above description has been written with the case of an arithmetical order involving the store in mind. In the case of other orders (e.g., left or



STAGE 2.

FIG. 2

right shift, or a conditional order), stage II is simplified in that the location section of the "order interpreter" is not used. For example, during stage II for a conditional order, the order gate is connected to the sequence control tank through the gate shown in the diagram and the whole order transferred from one tank to the other. When an order is in the sequence control tank, only the numerical part is used, the function part being irrelevant.

3. Initial Orders. From what has been said, and from an examination of the order code, it will be seen that the input mechanism is controlled by

program orders. Unless, therefore, there are some orders in the store at the beginning of the computation, nothing can be taken in through the input, and the machine cannot start. For this reason, there is a sequence of orders, known as initial orders, permanently wired onto a set of uniselectors (rotary telephone switches). These orders can be transferred to the store by pressing a button.

There is considerable latitude in the choice of the initial orders, although once they have been wired onto the uniselectors, it is not easy to change them. The initial orders used in the EDSAC at present enable orders punched in the following form to be taken in from the tape. First a letter indicating the function is punched, then the numerical part of the order in decimal form, and finally the letter F or D indicating, respectively, that the order refers to a long or a short number. If the order has no numerical part, it is punched simply as a letter followed by F. Under the control of the initial orders the machine converts the numerical part of the order to binary form and assembles the order with the function digits and the numerical digits in their correct relative positions.

M. V. WILKES
W. RENWICK

The University Mathematical Laboratory
Cambridge, England

¹ M. V. WILKES, "The design of a practical high-speed computing machine. The EDSAC," *R. Soc., London, Proc.*, v. 195, 1948, p. 274-279.

M. V. WILKES, "Programme design for a high-speed automatic calculating machine," *Jn. of Sci. Inst. and Phys. in Industry*, v. 26, 1949, p. 217-220.

M. V. WILKES & W. RENWICK, "An ultrasonic memory unit for the EDSAC," *Electronic Engineering*, v. 20, 1948, p. 208-213.

Convergence Rates of Iterative Treatments of Partial Differential Equations

1. Introduction. The development of high-speed digital computers¹ has made feasible the numerical solution by iterative methods of some partial differential equations. The convergence rates of several such iterative methods are estimated here. It is found that with the familiar elementary iterative methods some quite simple problems require prohibitive computational labor.

The iterative methods here considered are related to the various forms of the SOUTHWELL "relaxation method"^{2,4} in that they involve successively applied local corrections to improve an approximate solution. However, these iterative methods are routinized in conformity with the requirements of automatic computers while the relaxation method is flexible and depends in an essential way on the skill of its practitioners.

2. Reduction to Finite Difference Form. The iterative methods of successive approximation considered here are, like the relaxation method, not directly applicable to partial differential equations (and associated boundary conditions) but only to the finite difference approximations to them derived in the customary way.³ For example, the LAPLACE equations, $\Delta\phi = 0$, applicable within a region \mathcal{R} in the x, y plane may be approximated by the