



---

Soviet-era science, translated into English

# A NEW PRINCIPLE FOR CONSTRUCTING MEMORY DEVICES

1960

SovietRxiv

---

View the original and related papers at <https://sovietrxiv.org/items/ru-196001.69909>

Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.

**Abstract**

**Full Text**

**CYBERNETICS AND CONTROL THEORY**

**G. G. STETSURA**

**A NEW PRINCIPLE FOR CONSTRUCTING MEMORY DEVICES**

*(Presented by Academician A. A. Dorodnitsyn on 24 II 1960)*

This paper proposes a principle for constructing a parallel-action memory device that makes it possible to find rapidly the information recorded in it when what is specified is not the storage location, but an arbitrary part of the information itself, serving as the distinguishing feature for the search.

Let there be a memory array divided into separate equal units of information  $y_i$ , for the representation of each of which a set  $U_m$  of binary positions is allotted. The assignment of each particular  $y_i$  is equivalent to the determination of a subset  $Y_i$  in  $U_m$ , containing those binary positions which in the given  $y_i$  carry the binary unit.

The first search problem to be solved is the following. Among the set  $y_i$ ,  $i = 1, \dots, n$ , find those  $y_j$  which coincide with the query  $x$ .

Suppose that, along with  $y_i$ ,  $\bar{y}_i$  is stored—the information written in inverse code ( $Y_i \cup \bar{Y}_i = U_m$ ); and that, along with the query  $x$ , we specify  $\bar{x}$  ( $X \cup \bar{X} = U_m$ ). To carry out the search, perform the following operations:

1. Form  $\bar{X} \cap Y_i$ ,  $i = 1, \dots, n$ , and select those  $y_k$  and the corresponding  $\bar{y}_k$  for which  $\bar{X} \cap Y_k = 0$ , i.e.

$$Y_k \subseteq X. \tag{A}$$

2. For the selected information, form  $X \cap \bar{Y}_k$  and select those  $\bar{y}_j$  and the corresponding  $y_j$  for which  $X \cap \bar{Y}_j = 0$ , i.e.  $\bar{Y}_j \subseteq \bar{X}$ , or

$$Y_j \supseteq X. \tag{B}$$

The selected  $y_j$  satisfy (A) and (B); consequently  $y_j = x$ , i.e. the required information has been found.

Let us now introduce into consideration indifferent symbols such that  $X \cap \bar{Y}_i \equiv 0$ ;  $\bar{X} \cap Y_i \equiv 0$ , if the query consists of indifferent symbols. For such a query all positions in  $x$  and  $\bar{x}$  are zero. Now one can broaden the problem and search for those  $y_j$  which contain the query  $x$  within themselves. For this purpose, to that part of  $y_i$  which is immaterial for the search, we shall place in correspondence

in the query  $x$  indifferent symbols. If the location of the part of  $y_i$  that is essential for the query is unknown, then it is necessary to carry out a search for a sequence of queries sufficient to check the whole of  $y_i$ .

A memory device (M.D.) in which the described search principle is realized may be constructed as follows. Each

the pair  $(y_i; \bar{y}_i)$  is recorded on a separate information carrier (I.C.), which has one output and a number of inputs equal to twice the number of digits in  $U_m$  (since  $y_i$  and  $\bar{y}_i$  are stored), so that each digit in  $y_i$  and  $\bar{y}_i$  has its own input. The information on the I.C. is recorded by establishing a connection between the input and the output if the given digit in  $y_i$  contains 1, and by breaking the connection if the given digit in  $y_i$  contains 0. The information carriers corresponding to different  $(y_i; \bar{y}_i)$  are combined by inputs in such a way that all first inputs of  $y_i$ , all first inputs of  $\bar{y}_i$ , all second inputs of  $y_i$ , all second inputs of  $\bar{y}_i$ , etc., are connected. Thus we have  $1, 2, \dots, m$  common inputs of the  $y_i$ ' s and  $1', 2', \dots, m'$  common inputs of the  $\bar{y}_i$ ' s.

The query is posed by applying signals to the inputs of the array of information carriers. A signal is applied if, in the query, the given input corresponds to 1, and is not applied if it corresponds to 0. To the inputs  $1, 2, \dots, m$ , which correspond to the storage location of  $y_i$  on the information carriers, signals corresponding to  $\bar{x}$  are applied; to the inputs  $1', 2', \dots, m'$ , which correspond to the storage location of  $\bar{y}_i$  on the I.C., signals corresponding to  $x$  are applied. The I.C. must have the property that a signal arriving at an I.C. input connected to the output appears at the output. Then it is clear that, when signals from  $x$  and  $\bar{x}$  are applied simultaneously to the outputs of the I.C. array, signals will not appear only at the inputs of those I.C.' s whose information satisfies conditions (A) and (B), since only in this case will all query signals fall on broken connections and not reach the output. Consequently, the unexcited outputs will immediately indicate the storage locations of the desired information, and it can then be easily retrieved. If the connections are symmetric, then, in order to read the information found, it is necessary to remember the unexcited outputs and apply a readout signal to them in sequence; then at the input of the information array there will appear signals representing all the information from the  $(y_i; \bar{y}_i)$  being read at that moment. (Asymmetric connections can also easily be used.)

From the described search logic it is clear that the connections on the I.C. must not be destroyed when a query signal reaches them. This requirement is satisfied, for example, by transfluxors, which can be used to create random-access memory devices of the described type; the connection may also be implemented by means of capacitors, ferrite cores with a readout winding passed through them when a connection is present and not passed through when a connection is absent, if the device is not required to be random-access.

In a search, two principal cases may be encountered:

1. Only one  $y_i$  answers the query. Then there will be only one unexcited output of an I.C. In this case it is sufficient to have, at the output of each

I.C., an inverter which, after the necessary delay, can send a signal to the I.C. connected with it. Let us give two examples illustrating this search case.

- 1) All  $y_i$  are divided into two parts  $y'_i$  and  $y''_i$ . Each  $y'_i$  corresponds to its own number, recorded in  $y''_i$ . A number is specified; it is necessary to read the information  $y'_i$  corresponding to this number. A particular case of this type of search corresponds to the operation of an ordinary parallel-address random-access memory.
  - 2) The memory array is a dictionary. Each  $y_i$  consists of a word  $A$  and its translation. The positions of the words  $A$  in  $y_i$  are known. It is necessary, for a given word, to find its translation. In this case the answer is also unique, and the search time does not depend on the size of the dictionary.
2. Several  $y_i$  simultaneously answer the query. In this case, in order to read out all the information found, a sequential polling of the outputs of the S. D. is necessary. If one searches for  $y_j$  containing the information of the query at a previously known location, then a saving in time, as compared with a sequential examination of all the information, can be obtained only by a sequential-parallel examination of the outputs of the S. D., which is immeasurably simpler to carry out than a sequential-parallel examination of the  $y_i$ . An entirely different result is obtained when the location in  $y_i$  of the information answering the query is unknown. For example, it is required to find the  $y_j$  in which  $x_1 \wedge x_2 \wedge \dots \wedge x_k$  is contained, the location of each  $x_j$  in  $y_i$ ,  $i = 1, \dots, n$ , being arbitrary. In this case the inverters at the outputs of the S. D. must be replaced by more complex logical devices, which must select the  $y_j$  satisfying the conditions of the query. The search for information can then be performed considerably faster than with sequential examination, since the complex search is carried out simultaneously over the entire array of information, while the sequential search for marks indicating that information has been found is performed once and usually constitutes a negligibly small part of the whole search.

In order to illustrate the possibilities of the proposed S. D., let us give the solution of the problem of searching for  $y_j$  containing a numerical parameter in a specified range. For simplicity we shall assume that the location of the parameter in all  $y_j$  is one and the same and is known to us in advance. Suppose that, for representing the parameter, the decimal system is used with the following method of coding. To represent a decimal digit, 9 binary positions are allotted. For each digit, as many ones are entered in them as the value of the digit being represented: for 4—000001111, for 5—000011111, and so on. This code makes it possible most simply to separate digits greater than or equal to a given one from digits less than or equal to a given one. To find the  $y_j$  in which the parameter in a specified position contains a digit greater than a given  $z$  or equal to it, one may use the criterion

$$Z \cap \bar{Y}_i = 0.$$

An indifferent code is then fed to  $y_i$ . To find the  $y_i$  in which the parameter in a specified position contains a digit less than a given  $z$  or equal to it, one may use the criterion

$$\bar{Z} \cap Y_i = 0.$$

An indifferent code is then fed to  $\bar{y}_i$ . In both cases  $z$  is fed to the location of the specified position of the parameter in  $y_i$ . It is clear that fulfillment of both conditions will select the  $y_j$  containing a parameter in which the digit  $z$  stands exactly in the specified position. It is possible simultaneously to find the  $y_j$  whose parameter in the given position contains a digit greater than or equal to one of the specified digits  $z_1$  and less than or equal to another of the specified digits  $z_2$ . For this,  $y_i$  satisfying the conditions

$$Z_1 \cap \bar{Y}_i = 0; \quad \bar{Z}_2 \cap Y_i = 0.$$

are selected. Using these possibilities, the search for  $y_j$  with a parameter lying in a specified range can be carried out very quickly, as is illustrated by the following example. Find documents containing a parameter lying in the range from +25 to +73. For this purpose,  $y_j$  satisfying

to the following queries, submitted to the location of the parameter in  $(y_i; \bar{y}_i)$ .

Query $x$ , submitted to $\bar{y}_i$	Query $x$ , submitted to $\bar{y}_i$	Query $\bar{x}$ , submitted to $y_i$	Query $\bar{x}$ , submitted to $y_i$	Search result
tens	units	tens	units	Search result
00111111	00000000	11000000	111111000	from 73 to 70
000000111	00000000	11100000	00000000	from 69 to 30
000000011	000011111	11111100	00000000	from 29 to 25

It can be shown that, in order to find  $y_j$  containing a parameter in a range with  $N$ -digit decimal boundaries, a search is required for no more than  $2N$  queries of the type given in the example.

Received  
24 II 1960

*Note: Figure translations are in progress. See original paper for figures.*

*Source: Math-Net.Ru and CyberLeninka. Machine translation. Verify with the original.*