объединенный
институт
ядерных
исследований
дубна

E11-93-206

N.M.Nikityuk

CROSSBAR SWITCHES WITH ALGEBRAIC STRUCTURE FOR MULTIPROCESSOR SYSTEMS

Submitted to «Автоматика и телемеханика»

1993

1. INTRODUCTION

One of the most fundamental functions in data processing is data switching (or line switching). The role of this function is to essentially increase the importance in modern digital systems. It is well known that as the switching speeds of computer devices approach the limit, any further improvement of computer throughput can be processed simultaneously due to increasing the number of bits. In the present-day computers the communication function is often implemented by means of a switch (a bus switch or a crossbar switch). The cheapest switch is a time-shared bus. However, a time-shared bus has a very limited transfer rate which is inadequate for even a small number of processors. At the other end of the bandwidth spectrum is the full crossbar switch, which is also the most expensive switch. In a crossbar switch every input port can be connected with a free output port without blocking. A crossbar 2 × 2 switch is widely used for the construction of networks [1,2]. But much time and complicated control are required to construct large networks. As noted in [3—4], there are no switches with reasonable cost and performance which have prevented the growth of large multiprocessor systems. The switches described in this paper have the following principal differences from the well-known ones. Data bits are not switched from inputs to outputs. An input information is encoded in a code over linear operation, for example, AND operation is executed. This approach allows us to construct new economical and fast crossbar switches. Two methods are described: switches with the Hadamard matrix structure and switches over the Galois field GF (2^m).

The Hadamard matrix was widely used in the 50s and 60s for the construction of load-sharing matrix switches. The load-sharing switches combine the power from fast, low-power drivers into a fast, high-power memory driver used in a magnetic core memory [5,6]. A not full crossbar switch is described in [7]. The switch allows one to connect at a given time any pair of senders and receivers. A more detailed description of such switches is given in [8]. A transformation switch (key) was the main part on such switches. It is shown below that the use of modern methods of construction of logic devices along with the method of data encoding makes it possible to construct fast and economical crossbar switches.

2. CROSSBAR SWITCHES WITH THE HADAMARD MATRIX STRUCTURE

1. Some properties of the Hadamard matrix [9,10]. The hadamard matrix is an orthogonal matrix of the order n which elements are plus and minus ones. We will use the standard Hadamard matrix of even order n where plus and minus are changed by ones and zeros. The minimum order of the Hadamard matrix is equal to two. So, for $n = 2$ there are four rows

$$11, 10, 01 \text{ and } 00.$$

Two matrices of the second order can be constructed using these rows

$$H(2) = \begin{vmatrix} 11 \\ 10 \end{vmatrix}, \quad -H(2) = \begin{vmatrix} 00 \\ 01 \end{vmatrix}.$$

There is a simple rule for the construction of the Hadamard matrix of higher order. If we have the matrix $H(n)$, then the matrix $H(2n)$ obtained by interaction of the matrix $H(n)$ and the inversion matrix $-H(n)$

$$H(2n) = \begin{vmatrix} H(n) & H(n) \\ H(n) & -H(n) \end{vmatrix}$$

will be of order $2n$.
We will use for illustrating the following matrices which rows are numbered

$$H(8) = \begin{array}{c|c|} 1 & 11111111 \\ 2 & 10101010 \\ 3 & 11001100 \\ 4 & 10011001 \\ 5 & 11110000 \\ 6 & 10100101 \\ 7 & 11000011 \\ 8 & 10010110 \end{array} \quad \text{and} \quad -H(8) = \begin{array}{c|c|} 1 & 00000000 \\ 2 & 01010101 \\ 3 & 11001100 \\ 4 & 01100110 \\ 5 & 00001111 \\ 6 & 01010101 \\ 7 & 00111100 \\ 8 & 01101001 \end{array}.$$

It is important that the Hadamard matrix of order n forms the words of the error correcting codes consisting of n symbols a minimum coding distance $n/2$. From the practical viewpoint this means that the Hadamard code made from the matrix of order 8 has a coding distance of $d = 4$. Such a code can correct one and detect multiple mistakes according to the relation

$$t = \frac{d-1}{2},$$

where t is the number of correcting mistakes. Besides the coding distance d grows linearly with increasing the number of n. The matrix $H(4)$ has a minimum coding distance of $d = 2$ for the Hadamard code. It should be noted that the coding words composing the rows of the matrix — $H(n)$ can be obtained in another way. It is enough to take a digit by a digit modulo 2 two all possible rows of the matrix $H(n)$. For example, we get the first row of the matrix $-H(8)$ by adding modulo 2 to the first row of the matrix $H(8)$. If we add the first and second rows of the matrix $H(8)$, we get the second row of the matrix $-H(8)$. Below we show how the matrices $H(8)$ and $-H(8)$ can be used for the construction of crossbar switches without blocking and correcting possibility.

2. Crossbar switches for nxn directions. The idea of construction of such switches is very simple. Every n bit of input data obtained from n senders is encoded to the n-bit Hadramard code which is then decoded. The outputs of a decoder are the outputs of one unit of a one-bit switch. In other words, Hadamard codes take the information on the number of active receivers. A typical scheme of the one-bit nxn switch includes AND-gates and a decoder which decodes the n-bit Hadamar code to the unitary code. As an example, fig.1 gives a block-diagram of the first unit 8×8 switch. The address is given to switch a signal from the first sender to the second receiver. The input of the switch is connected to the first group of the inputs of the AND gates. The second group of the AND gates is connected to logical ones and to logical zeros according to the second row of the $H(8)$ matrix. Thus, the second group of the inputs of the AND gates is used as address receivers. A one-bit crossbar switch without blocking with correcting capabilities can be constructed using n analogous units. For this purpose corresponding outputs of the decoders must be connected by an OR-gate. Moreover, it is necessary to foresee both correcting and detecting multiple mistakes which can arise by switching. Every decoder has eight inputs and eight outputs. It takes the function of decoding the Hadamard code to the position unitary code taking into account mistakes which can arise by communication.

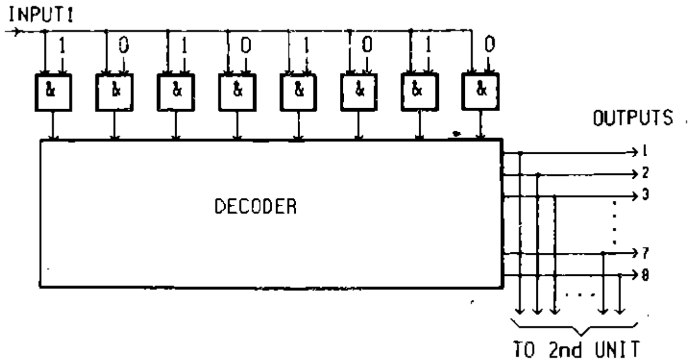


Fig.1. Scheme for one unit of the crossbar switch with Hadamard matrix structure of the 8th order. & — AND-gates

For example, a signal at the first output of a decoder must appear not only when it is sent to its inputs but also when there appear both code 11111111 and the codes with mistakes: 01111111, 10111111, 11011111 and so on. Parallel with correcting ability, the crossbar switch, described here, has a distinguishing feature — the absence of triggers in switch logic. As a result, synchropulses are not required. Synchropulses are needed only for strobing the output register.

It is necessary to choose the Hadamard matrix of higher orders n for the construction of switches for a larger number n. However, the coding distance increases linearly with increasing the number n. It is clear that the correcting ability decreases, and the abundance of logical schemes is needed.

3. Crossbar switch with $(2n - 1)$ *inputs.* Another approach is to increase the number of inputs (outputs) of a crossbar switch to $(2n - 1)$ without the abbundance of logical devices. A rectangular Hadamard matrix having $(2n - 1)$ rows and n columns can be constructed using the matrices $H(n)$ and $-H(n - 1)$ obtained by deleting the first (zero) row

$$H(2n - 1) = \begin{vmatrix} H(n) \\ -H(n-1) \end{vmatrix} .$$

It is clear that the delay T_d of Hadamard matrix structure switches is small and does not exceed three delays of the AND gate.

3. CROSSBAR SWITCHES OVER GALOIS FIELD $GF(2^m)$

In this section we consider the method of construction of switches which is based on the properties of Galois field $GF(2^m)$ widely used for the construction of effective codes correcting to $n \leq n/2$ mistakes in n-binary words (BCH-codes). Other areas of application of finite fields are known. The author has shown that by using the algebraic theory BCH-codes one can construct coordinate processors and majority coincidence circuits for large inputs [11,12].

In order to make this article easier for the readers who are not familiar with the rules of execution of operations over $GF(2^m)$ elements, necessary information on these rules is presented below which has no claim on completness and generality. Concrete examples are given for $m = 4$.

As is known, Boolean algebra is the theoretical base of modern computer technology where the discrete function is equal to 0 or 1 in the binary case. As is shown in [11,12], Boolean algebra is the particular case of Galois algebra. Operations in Galois algebra are determined over the main field $GF(2)$ having two elements 0 and 1 and the extended field to m-th power having $2^m - 1$ nonzero elements which are considered as m-bits words of cyclic codes. The number of nonzero elements of the finite field $GF(p^m)$ is equal to some power of its characteristics, i.e., $n = 2^m - 1$. The number of different elements of the field is called its order. All elements of the finite field can be obtained by means of irreducible polynomials which tables are presented in [13]. We have chosen the irreducible polynomial of the 4th degree $F(X) = X^4 + X + 1$ $(m = 4)$. It should be noted that sign «+» will be used to denote modulo 2 additions. For $m = 4$ the number of nonzero elements equals 15. Among this elements there are four linearly independent (basis) elements: $a^0 = 1000$, $a^1 = 0100$, $a^2 = 0010$ and $a^3 = 0001$. One of these elements, a^1, is the root of the polynomial $F(X)$. Then each nonzero element can be presented as a degree of element a^1. This means that the multiplicative group of the finite field is cyclic in character. The least positive number n, for which $a^n = a^0 = 1$, is referred to as the order of element a^1. If the order of element a^1 equals n, elements $a^0, a^1, a^2 \ldots a^{n-2}, a^{n-1}$ are different. Thus, in our example $n = 15$. Therefore, e.g., $(a^{30}) = (a^{15})^2 = a^0$; $(a^{31}) = a^{15}a^{16} = a^1$, etc. Taking into account that a^1 is the root of the polynomial $F(X)$, the remaining elements of $GF(2^4)$ can be obtained from the equation $a^4 + a^1 + + 1 = 0$, i.e. $a^4 = a^1 + 1 = 1100$; $a^2 + a^1 = a^5 = 0110$ and so on. All of 15 nonzero elements are listed in the table.

Fig.2 gives the table of multiplication of two elements A and B by moludo of the irreducible polynomial $X^4 + X + 1$. As is shown below, this table serves to get receiver addresses.

It is also convenient to present the field elements A and B as polynomial of degree $m - 1$. For $m = 4$ we have: $A = A_0a^0 + A_1a^1 + A_2a^2 + A_3a^3$ and $B = = B_0a^0 + B_1a^1 + B_0a^2 + B_0a^3$, where $A_0 = A_3$ and $B_0 = B_3$ are equal to 0 or 1. Thus, for the element $A = a^8$ $A_0 = A_2 = 1$ and $A_1 = A_3 = 0$. In the execution of operations with the Galois field elements as in conventional binary arithmetic, there are some differences in the rules of calculation carried out «manually» and by a computer. This concerns such operations as multiplication, raising to power, division and extraction of the square root. It is simple to perform multiplication and division operations manually: the degree of the product of elements equals the sum of the degree of factors. In this case the degrees are summed modulo

Table. The elements of Galois field GF(2⁴)

$a^0 = 1000$
$a^1 = 0100$
$a^2 = 0010$
$a^3 = 0001$
$a^4 = 1100$
$a^5 = 0110$
$a^6 = 0011$
$a^7 = 1101$
$a^8 = 1010$
$a^9 = 0101$
$a^{10} = 1110$
$a^{11} = 0111$
$a^{12} = 1111$
$a^{13} = 1011$
$a^{14} = 1001$
$a^{15} = 1000 = 1$

A →

B x	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
↓ a^0	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}
a^1	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0
a^2	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1
a^3	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2
a^4	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3
a^5	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4
a^6	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5
a^7	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6
a^8	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7
a^9	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8
a^{10}	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9
a^{11}	a^{11}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}
a^{12}	a^{12}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}
a^{13}	a^{13}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}
a^{14}	a^{14}	a^0	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}

Fig. 2. Table for multiplication of two elements over Galois field $GF(2^4)$

2^m. The operation of division of elements A by B is equivalent to that of multiplication of element A by inverse element B^{-1}. Inverse element B^{-1} of element B is determined from the condition: $B \times B^{-1} = 1 = a^0$. Thus, for element a^{12} element a^3 is an inverse element as $a^{12}a^3 = 1$. Addition and subtraction operations in the field $GF(2^m)$ are equivalent and carried out by modulo two. The operation of multiplication of two elements is carried out in the same manner as that of ordinary numbers with the difference that this operation is performed by modulo irreducible polynomial. We get the following Boolean expressions for two elements A and B by direct multiplication of the polynomials

$$P_0 = A_0B_0 + A_1B_3 + A_2B_2 + A_3B_1$$

$$P_1 = A_0B_1 + A_1B_0 + A_1B_3 + A_2B_2 + A_2B_3 + A_3B_2 + A_3B_1$$

$$P_2 = A_0B_2 + A_1B_1 + A_2B_0 + A_2B_3 + A_3B_2 + A_3B_3$$

$$P_3 = A_0B_3 + A_1B_2 + A_2B_1 + A_3B_0 + A_3B_3$$

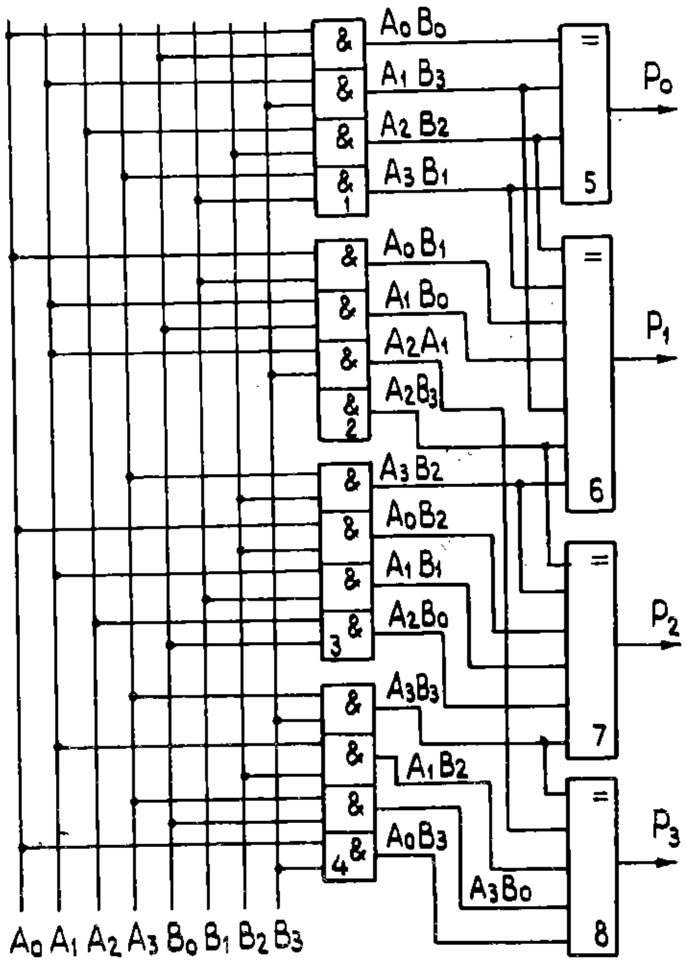


Fig.3. Principal scheme for multiplication of two elements over Galois field $GF(2^4)$. Microcircuits: 1—4 — MC 10102; 5—8 — MC10160

It is obvious that the product is zero if one of the multipliers is equal to zero. Fig.3 gives a principal scheme for multiplication of two elements over Galois field $GF(2^4)$. The scheme is very simple. It is also important that the number of logical gates necessary for the construction of multipliers for large m increases linearly, and the delay time remains constant.

4. Scheme of the one-bit nxn crossbar switch. The method for switch construction is of the following way. Input bits are arriving from n senders presented as unitary code n-bit words which is encoded to Galois field elements in increasing order of their degrees. The outputs of the encoders are connected to the first group of the inputs of the multipliers in Galois field $GF(2^m)$. The addresses of the receivers are supplied to the second group of inputs according to the table (fig.2). Then the results of multiplication are decoded. Each decoder *has m inputs and* 2^m outputs. And, besides, the corresponding outputs of the decoders are connected in case of N-bit switch. For example, to send one bit from input 2 to receiver 5, it is necessary to deliver the control word a^7. We get $a^2a^7 = a^9 = = 0101_2$ at the outputs multiplier. As a result, we get a pulse corresponding to a logical one at 5th output of the binary decoder. Figure 4 gives a typical scheme of one (first in order) unit. It is clear that the scheme of the encoders is very simple. They are the sources of logical one and logical zero. The delay T_c of a switch can be calculated from the relation

$$T_c = 3T_\& + T_\Sigma,$$

where $T_\&$ is the delay of the AND-gate which is in the multiplier and decoder and T_Σ is the delay of a modulo 2 adder which is equal to two delayes of the AND-gate. Then the total delay is $4T_\&$. It should be noted that two approaches can be suggested to construct a switch having inputs (outputs) which are multiple to a degree of two as Galois field $GF(2^m)$ has 2^m— 1 nonzero elements. 1) To use the modular nature of the Galois field; 2) to decide this problem by using complementary scheme in the given Galois field. For example, to construct a one-bit 16×16 switch, the $GF(2^5)$ field can be used instead of the Galois field $GF(2^4)$ and multiplier can be minimized. It is clear, the switches over Galois field $GF(2^m)$ do not correct mistakes. However they are more economical in comparison with switches having the Hadamar matrix structure. The number of control bits needed for N-bit nxn switch is equal to Nxn. One can recommend the following next irreducible polynomials for the construction of switches for $n = 31, 63, 127, 511$ and 1023: $X^5 + X^2 + X + 1$; $X^6 + X + 1$; $X^6 + X + 1$; $X^8 + X^4 + X^3 + X^2 + 1$; $X^9 + X^4 + 1$ and $X^{10} + X^3 + 1$.

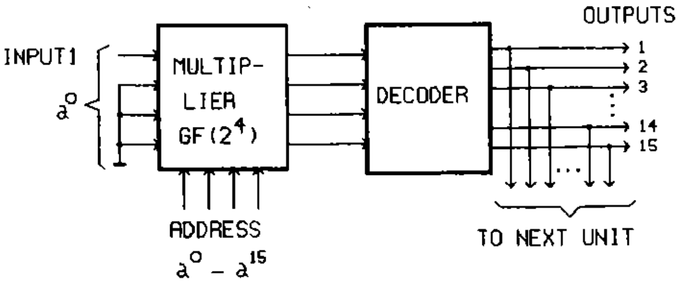


Fig.4. Block-diagram for one unit of the crossbar 15×15 switch over Galois field $GF(2^4)$

5. *Synchroneous switch.* In practice, we meet with a simpler problem when it is necessary to switch an identical data from detectors having different delays. The essence of the problem is explained in fig.5 for $n = 7$. Information is sent through processors and FIFO-memories where data are preliminarily processed and delayed. Besides, events $A1-A7$ must be transmitted to the receiver A, events $B1-B7$ to the receiver B and so on. For this task it is enough to use one unit of the switch (fig.6). Besides input symbols are sent through the AND-gates which are opened with the help of a decoder. Synchropulses are required both for the execution of this operation and for changing the addresses senders and receivers [17]. The encoder over $GF(2^4)$ consists of four OR-gates and executes the encoding of the unitary position code to the elements over Galois field $GF(2^4)$. Let us consider the operation of the switch. One-bit data from 15 sources arrive to the first group of the inputs of AND gates &1–&15. The outputs of the decoder are connected to the second group of the inputs of the gates. We get the source addresses either in the binary code or in the $GF(2^4)$ elements and the receiver addresses in the $GF(2^4)$ elements. Suppose that it is necessary to switch a pulse from the second source to the sixth receiver. In this case gate 2 is open and the element $a^1 = 0100$ appears at the outputs of the encoder E.

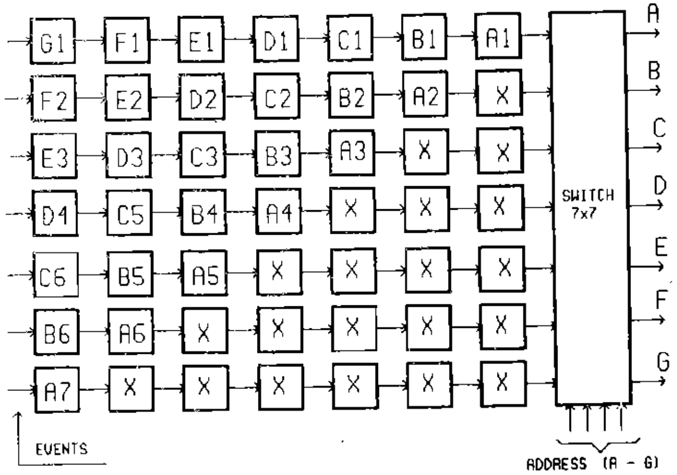
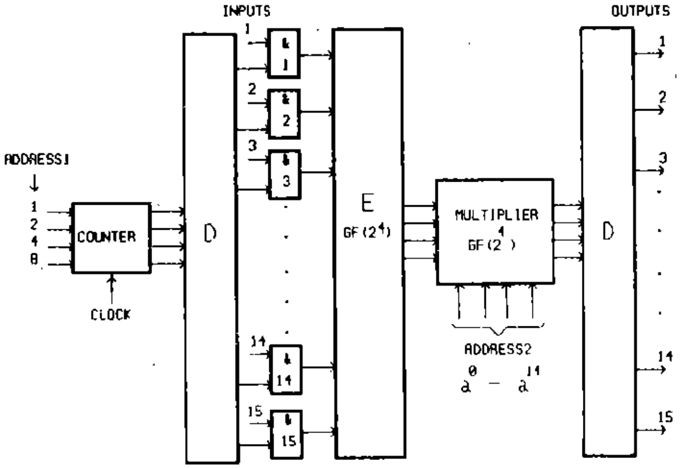


Fig.5. A switching data on events obtained from detectors having different delays. $A-G$ receivers; $A1-A7,B1-B7$, etc. — information on events; X — arbitrary information



Fgi.6. Block-diagram of the one-bit *nxn* synchronous switch. ADDRESS — addresses of sources; ADDRESS2 — addresses of the receivers; E — encoder; D — decoder.

Simultaneously the element $a^3 = 0001$ is generated at the second group of the inputs of the multiplier. After multiplication we obtain $a^4 = 1100$ which it considered as a usual binary number $1100_2 = 6$. As a result a pulse is formed at the sixth output of the decoder. The matrix of triggers is used in switch for this purpose [18].

It should be noted that a switch with a correcting possibility of $t \leq n/2$ can be constructed over Galois field $GF(2^m)$ if we choose code words of the BCH-code (as it takes place in switches with the Hadamard matrix structure).

CONCLUSION

The main difference of the switches described in this paper is that not information but addresses of an active source are switched. As a result, these switches have a simple modulo structure, and only a combination logic is needed for their realization. The absence of a bus which requires high power current switches, the correcting possibility, small delays and advanced technology make it possible to construct effective switches with many inputs for multiprocessor systems and high-speed networks, communication and scientific instruments with wide functional possibilities.

REFERENCES

1. Feng T.Y., Wu C.L. — IEEE Trans. on Computers, 1980, v.C-30, No.10, p.7433.
2. Booth T.L. — IEEE Trans. on Computers, 1968, v.C-17, No.5, p.452.
3. Jansen P.G., Kessels J.L.W. — IEEE Trans. on Computers, 1980, v.C-29, No.10, p.884.
4. Veselovskiy G.G., Karavai M.F., Kuznechik S.M. — Avtomatika i Telemehanika, 1989, No.2, p.3.
5. Constantine G.A. — IBM J. Res. Dev., 1958, v.2 July, p.204.
6. Chien R.T. — IBM J. Res. Dev., 1960, v.4. October, p.414.
7. Semakov N.V. — Combinatorial Transformation Switch. Byulleten Izobreteniy i Tovarnykh Znakov. 1964, No.21, p.49. Avtorskoe svidetelstvo No.166165, Cl. G 06 F 42m, 1402.
8. Nikityuk N.M. — In: Proc. 6 Conf. of JINR of Nuclear Electronics, Warsaw, 1971, D13-6210. Dubna, 1972, p.234.
9. Levenshteyn V.I. — Problemy Kibernetiki, 1961, v.5, p.123.
10. Nikanorof A.A. — Problemy peredachi informatsii, 1974, v.X, No.2, p.95.
11. Pradham D.K. — IEEE Trans. on Comput., 1978, v.C-27, No.3, p.239.
12. Benjauthrit B., Reed I. — IEEE Trans. on Computers. 1976, v.C-25, No.1, p.78.
13. Peterson Y. — Error-Correcting Codes, 1964, «Mir», Moscow, p.290.
14. Nikityuk N.M. — Switch for Multiprocessor Systems over Galois Field *GF*(2^m). Byulleten Otkrytiy i Izobreteniy, 1983, No.44, p.186. Avtorskoe svidetelstvo No.1057951, Cl. G. 06 f 15/16.
15. Dedyulin K.A. et al. — Elektronnaya Promyshlennost, 1970, v.134, No.6, p.14.

Received by Publishing Department
on June 3, 1993.