

UDC 004.724.2 + 004.272.43

 10.25209/2079-3316-2025-16-5-3-42


Group Computing in Non-blocking System Area Networks

Viktor Sergeevich **Podlazov**

V. A. Trapeznikov Institute of Control Sciences of RAS, Moscow, Russia

 podlazov@ipu.ru

Abstract. This paper is devoted to non-blocking optoelectronic networks in which group operations over a set of numbers are executed during the transmission of one number. The execution of group operations by photonic units placed at each network user is considered, and the circuitry of these units is presented. The possibility of executing group operations in cyclic sequences of nodes defined by Hamilton graphs is investigated. Procedures for constructing such cyclic sequences and executing group operations in them are proposed. The possibility of designing a non-blocking electronic system area network for any number of users is studied. A method for designing an optoelectronic network as a tandem network with optoelectronic switches controlled by their analogs from the electronic network is described. (*Linked article texts in English and in Russian*).

Key words and phrases: non-blocking system area networks, group operations, photonic and optoelectronic switches, Hamilton loops of network nodes, non-blocking tandem networks

2020 *Mathematics Subject Classification:* 65Y05; 68Q10

For citation: Viktor S. Podlazov. *Group Computing in Non-blocking System Area Networks*. Program Systems: Theory and Applications, 2025, **16**:5(68), pp. 3–42. (*In English, in Russian*). https://psta.psir.ru/read/psta2025_5_3-42.pdf

Introduction

In modern computing systems based on system area networks (SANs) being developed, data are transmitted together with commands, forming spatial data flow structures [1]. A particular case is computing systems with parallel data processing during their transmission in the network: a group operation over a set of numbers is executed while one number (the result of this operation) is transmitted via a common channel of the network. Such operations, called common channel computing (CCC) [2], can be executed in direct channels without intermediate buffering (on-the-fly). The simplest examples of CCC operations are summing a set of numbers from different nodes or finding the maximum of them.

A channel is common if users can transmit signals into this channel in parallel with their overlapping, which can be used for group computations. A common channel is direct if it has only transmission delays of signals or their combination delays, without any clock delays. CCC operations can be executed in cyclic sequences of network nodes forming Hamilton graphs. In this case, CCC operations on non-intersecting graphs can be executed in parallel.

Each network node needs executive units for different CCC operations. They can be built in the element base of optical channel splitters in a purely photonic implementation (without electronic control of units) [3]. By assumption, first, network nodes have the capability to transmit/receive data to/from executive units by optical signals; second, executive units work only with these optical signals. Such units allow executing arithmetic and logical operations over numbers represented by optical signals without using special photonic arithmetic-logical units (ALUs).

Cyclic sequences of nodes with direct channels can be constructed in non-blocking SANs, which are conflict-free on arbitrary permutations of packets between nodes. In the element base of switches and channel splitters (multiplexers/demultiplexers), it is possible to design non-blocking SANs with direct channels of any size according to the number of network users [4, 5].

Based on the above properties, the objective of this paper is to develop a method for designing a non-blocking SAN that executes CCC operations on-the-fly over an arbitrary set of nodes under a photonic implementation of executive units.

The following tasks are solved to achieve the objective:

- (1) selecting an appropriate method to change the values of bits in the channel based on their two-rail representation;
- (2) selecting an appropriate element base for photonic executive units;
- (3) developing basic elements for the photonic implementation of arbitrary logical functions;
- (4) synchronizing the bitwise overlapping of numbers in a common channel;
- (5) developing photonic units to execute CCC operations with numbers from different network nodes;
- (6) developing a method to design a non-blocking SAN of arbitrary size in the element base of switches and channel splitters of a given size;
- (7) developing a method to execute group operations in a SAN by forming Hamilton loops of network nodes;
- (8) developing a method to design a non-blocking photonic network as part of a tandem network in which photonic nodes are controlled by electronic nodes.

In Section 1, we present a method for building photonic CCC units and a circuitry for their implementation. Section 2 considers a method for designing non-blocking SANs with direct channels and forming Hamilton graphs in them, including a method for embedding CCC units. In Section 3, we describe a method for designing non-blocking optoelectronic networks with Mach–Zehnder interferometers as an application base for photonic executive units.

1. Distributed Arithmetic Based on Photonic Switches

To execute CCC operations, logical variables are transmitted via the channel in the two-rail form (along two lines, 0 and 1). Value 0 is transmitted by a signal in line 0 in the absence of a signal in line 1, and value 1 is transmitted by a signal in line 1 in the absence of a signal in line 0. Lines 0 and 1 pass through the converter C_i of each network node i . Each converter executes the unary operation O_i of transforming an input logical variable x_i to an output logical variable $y_i = O_i x_i$.

The operation O_i consists of the following transformations: repetition $y_i = x_i$ (“—”), negation $y_i = \neg x_i$ (“¬”), change into the value $y_i = 0$ (“0”), and change into the value $y_i = 1$ (“1”).

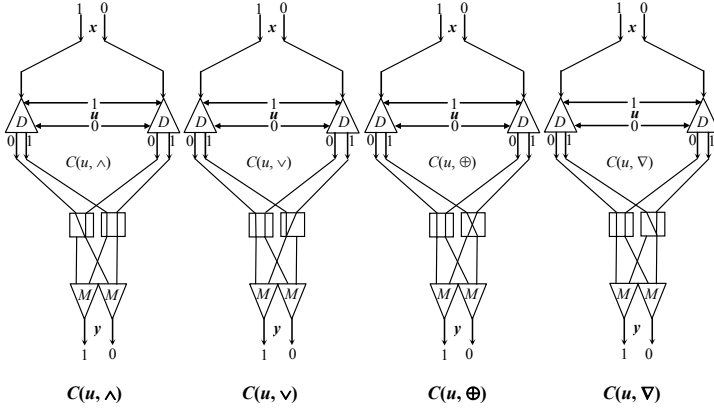


FIGURE 1. The circuits of logical functions from Table 1 for two-rail variables

TABLE 1. Logical functions by converter states for the adder

AND (logical conjunction)				OR(logical disjunction)				XOR(addition mod 2)				Initial Run			
$\aleph = \wedge$				$\aleph = \vee$				$\aleph = \oplus$				$\aleph = \nabla$			
x	u	O	y	x	u	O	y	x	u	O	y	x	u	O	y
0	0	«0»	0	0	0	«=»	0	0	0	«=»	0	0	0	«=»	0
0	1	«=»	0	0	1	«1»	1	0	1	«¬»	1	0	1	«0»	0
1	0	«0»	0	1	0	«=»	1	1	0	«=»	1	1	0	«=»	1
1	1	«=»	1	1	1	«1»	1	1	1	«¬»	0	1	1	«0»	0

The converter is used in four fixed states for which the operation O_i is considered to be executed. Each state is defined by some configuration of a pair of fiber sections, given by the value of the two-rail control variable u_i .

The circuits in Figure 1 show the use of such pairs of converters for implementing some logical functions on photonic demultiplexers D and multiplexers M for the two-rail variables x , u , and y .

For such circuits, we will follow the notation given at the bottom of Figure 1.

A photonic 1×2 demultiplexer D and a photonic 2×1 multiplexer M are used in Figure 1. They are assembled from several thin-film layers of metamaterials with electro-optical or magneto-optical properties [6, 7]. A peculiarity of these elements is that an optical signal is used to control switching. In the demultiplexer, they are converted into two control signals

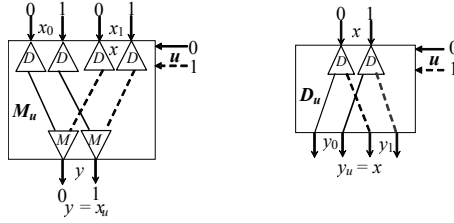


FIGURE 2. The circuits of a multiplexer and a demultiplexer for two-rail variables

of an electronic or magnetic nature and supplied to a guide layer, which specifies two propagation directions for the information signal received at its optical input. As a result, the input optical signal is sent to one of the two optical outputs.

In this paper, one two-rail control variable is used; therefore, other single-layer devices with two outputs (demultiplexer) or inputs (multiplexer), such as gallium arsenide films [8], can be selected. They allow building photonic logical elements directly controlled by light.

Table 1 provides the truth tables of the logical operations used in the circuits in Figure 1. Obviously, the operation $\aleph$ is defined only by the pair of states of the converter C_i , which are selected by the value of the variable u_i . As a result, it is possible to execute the binary operation $y_i = x_i \aleph u_i$ (any of the sixteen logical operations) on a pair of the converters. Figure 2 also shows the circuits of a multiplexer (M_u) and demultiplexer (D_u) controlled by the variable u for two-rail variables.

When nodes are placed sequentially, it is possible to sum numbers at the nodes or select the maximum of them during the transmission of one number via the common channel. For this purpose, the numbers are transmitted as binary ones sequentially bitwise. To execute their addition, the numbers are transmitted with the least significant bits first; to find their maximum, with the most significant bits first.

In non-blocking SANs, there always exist cyclic sequences of nodes in the form of Hamilton graphs, and each such graph has a path passing through each node exactly once. This property directly follows from those of non-blocking networks ensuring conflict-free routing on arbitrary permutations of data packets. With the use of Hamilton loops, conflict-free data transmission is guaranteed and linear sequences of photonic units implementing group CCC operations are embedded in non-blocking networks. In this case, all data sources and sinks are defined.

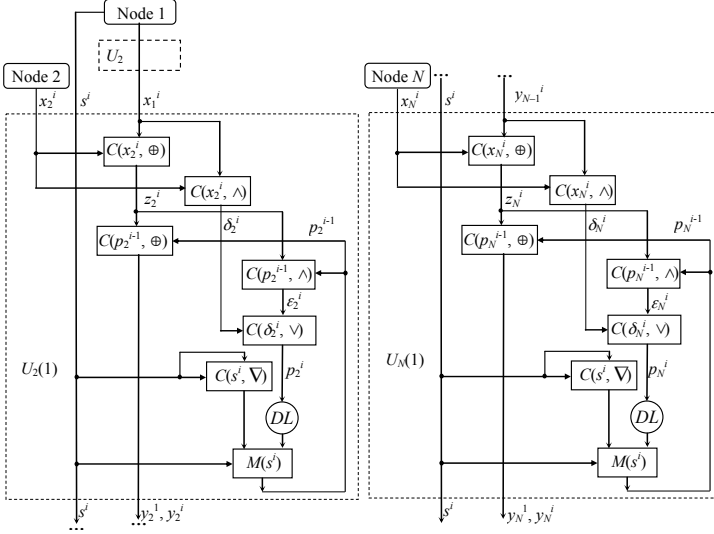


FIGURE 3. The N-node distributed adder

In each loop of nodes, an initial (first) node is separated to start the construction of the others (see the next section). This node triggers the command of the operation to be executed. For all nodes, the command sequentially synchronizes the start of executing a distributed operation r . At each node, there is a set of executive units, $U(r)$, configured to execute individual operations. The operation command selects a particular executive unit. Each executive unit consists of circuits implementing the necessary logical operations in a definite sequence specified by their interconnections.

Figure 3 shows the units $U_2(1)$ and $U_N(1)$ at the second and N th nodes of the cyclic sequence that sum N numbers.

The initial first node of the network does not use its unit $U_1(1)$. It outputs slot X_1 , containing a binary multi-bit number starting from the least significant bits (on the two-rail value line), and slot S of the same bit length (on the two-rail synchronization line). In slot X_1 , the bits are transmitted by the variables x_1^i ; in slot S , by the variables s^i , where i denotes the bit number. The first bit of slot S has value $s^1 = 1$ and the other bits $s^i = 0$ for $i > 1$. The dots on the lines indicate the nodes of light signal separation by branching optical lines. The unit $U_2(1)$ of the distributed adder is connected to the first and second network nodes.

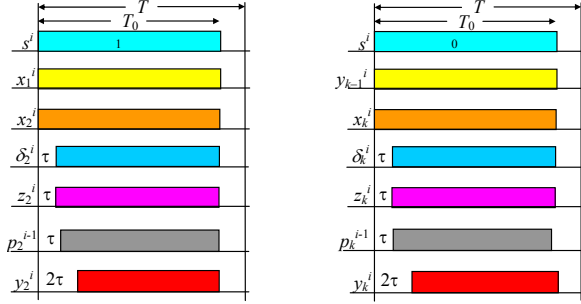


FIGURE 4. The timing diagram of the distributed adder

The circuits in Figure 3 are based on the well-known single-bit adder [9], supplemented with a circuit to form the zero carry value from the previous bit for the first bit and a circuit to carry the value to the next bit through a one-bit delay line DL .

Slots S , X_1 , and X_2 are supplied to the inputs of $U_2(1)$ synchronously bitwise; its output is slot Y_2 , containing the sum of numbers from slots X_1 and X_2 . In this unit, the values of the variables x_1^i and x_2^i are added mod 2 at the converter $C(x_2^i, \oplus)$ to produce an intermediate result, i.e., the variable z_2^i . Together with the carry p_2^{i-1} from the previous bit, the latter variable forms the output variable y_2^i of this bit at the converter $C(p_2^{i-1}, \oplus)$.

At the converter $C(x_2^i, \wedge)$, an intermediate variable δ_2^i is obtained from the variables x_1^i and x_2^i . It is further used at the converter $C(\delta_2^i, \wedge)$ to form the carry value p_2^i to the next bit. In turn, the last bit is generated from the intermediate variable ε_2^i obtained at the converter $C(p_2^{i-1}, \wedge)$ from the result z_2^i and the carry value p_2^{i-1} from the previous bit.

The circuit $C(s^i, \nabla)$ forms the carry value from the previous bit for the first bit. The carry value in the first and other bits is passed through the multiplexer $M(s^i)$.

Any k th node of the network ($2 < k \leq N$) transmits to its unit $U_k(1)$ slot X_k , containing the values of the variables x_k^i for each bit; the other input of this unit receives slot Y_{k-1} , containing the output bits y_{k-1}^i from the unit $U_{k-1}(1)$. In addition, this unit receives slot S from the executive unit of the previous node.

Figure 4 presents the timing diagram of the unit $U_2(1)$ in the first two bits with the following notations: T is the bit period; T_0 is the duration of active signals in it; τ is the time to tune the converter or multiplexer by the control variable u . Clearly, the output variable y_2^i appears in all bits

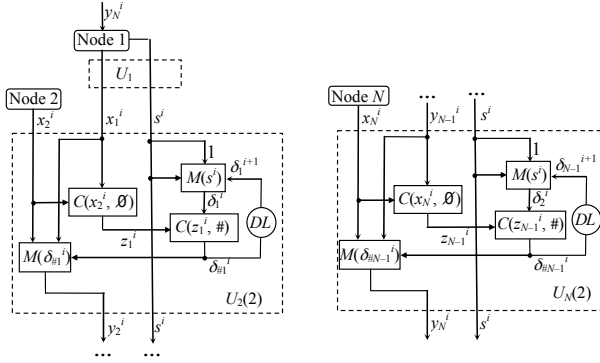
FIGURE 5. The N -node distributed maximizer

TABLE 2. Logical functions by converter states for the maximizer

Logical switching for the unit $U_k(2) (k \geq 2)$				Logical repetition for the unit $U_k(2) (k \geq 2)$			
$\aleph = \emptyset$				$\aleph = \# = \wedge$			
x_{k-1}	x_k	O	z_{k-1}	z_{k-1}	δ_{k-1}	O	$\delta_{\#(k-1)}$
0	0	«1»	1	0	0	«0»	0
0	1	«=»	0	0	1	«=»	0
1	0	«1»	1	1	0	«0»	0
1	1	«=»	1	1	1	«=»	1

with a delay of 2τ and is defined by signals of duration $T_0 - 2\tau$.

Also, Figure 4 (left and right) shows the timing diagram of the units $U_2(1)$ and $U_k(1)$ of the distributed adder (in the same notation). Note that the right diagram is valid for any unit $U_k(1)$, $k > 2$. In other words, the output variable y_k^i appears with a delay of 2τ with respect to the beginning of the synchronization signals s^i for any k and i (for any number of nodes and in all bits) and is given by signals of duration $T_0 - 2\tau$.

Along with the above CCC addition, the first node can also set the maximization operation (find the largest number of those placed at the network nodes). Following this command from the source node, an executive unit $U_k(2)$ (maximizer) is selected at each next node; see the circuit in Figure 5.

The units $U_k(2)$ also include circuits executing the logical functions from Table 2.

The first node of the network outputs to the unit $U_2(2)$ slot X_1 , containing a binary multi-bit number starting from the most significant

bits (on the two-rail value line), and slot S of the same bit length (on the two-rail synchronization line). In slot X_1 , the bits are transmitted by the variables x_1^i ; in slot S , by the variables s^i , where i denotes the bit number. The first bit of slot S has value 1 ($s^1 = 1$) and the other bits are 0 ($s^i = 0$ for $i > 1$).

Any other node of the network ($1 < k \leq N$) transmits to its unit $U_k(2)$ slot X_k , containing the values of the variables x_k^i for each bit (on the value line). In addition, this unit receives slot S from the executive unit of the previous node.

Figure 5 (left) shows the circuit of the unit $U_2(2)$ connected to the first and second network nodes of the distributed maximizer. The dots indicate the nodes of light signal separation by branching optical lines. Optical signals can be attenuated using photonic amplifiers at the output of the units $U_k(2)$ [10].

The variable δ_1^i must have the value $\delta_1^i = 1$ for all bits $i < j$ with $x_1^i \geq x_2^i$ and the value $\delta_1^i = 0$ for all bits $i > j$ (after the j th bit) with $x_1^j < x_2^j$. This value sequence of δ_1 is formed by the converters $C(x_1^i, \emptyset)$ and $C(z_1^i, \#)$ together with the multiplexer $M(s^i)$ and the one-bit delay line DL . The former converter generates the value of the intermediate variable z_1^i from those of the variables x_1^i and x_2^i , implementing logical switching (Table 2) for $k \geq 2$. The latter converter forms the value of the variable δ_1^i , implementing logical repetition (Table 2) for $k \geq 2$. The multiplexer $M(s^i)$ and the delay line DL produce the values of the variable $\delta_1^{i+1} = \delta_{\#1}^i$ for the next bit.

For bits $j < j$ with the value $\delta_1^i = 1$ stored, the output variable $y_2^i = x_1^i$ is obtained through the multiplexer $M(\delta_{\#1}^i)$. However, for bits $j > j$ with the value $\delta_1^i = 0$ stored, the output variable $y_2^i = x_2^i$ is obtained through the multiplexer $M(\delta_{\#1}^i)$. For the i th bit, the resulting value is $y_2^i = x_2^j = 1$.

Figure 5 (right) presents the circuit of the unit $U_k(2)$, actually repeating for ($k > 1$) that of the unit $U_2(2)$ with x_1 replaced by y_{k-1} and 2 by k in the subscripts.

Figure 6 shows the timing diagram of forming different variables for the j th and $(j + 1)$ th bits with the following notation: τ is the tuning time of the converters and multiplexer; T and T_0 are the durations of the bit and the active signal in it. Obviously, the active signal is shortened by a time not exceeding 3τ , which is independent of j .

Concluding section 1, note that the distributed photonic adder and maximizer can also be easily implemented in the electronic element base by replacing the delay line DL with a 1-bit shift register.

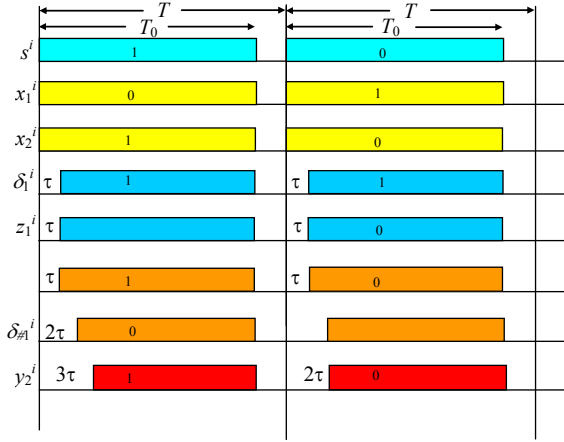


FIGURE 6. The timing diagram of the two-node maximizer for the j th bit

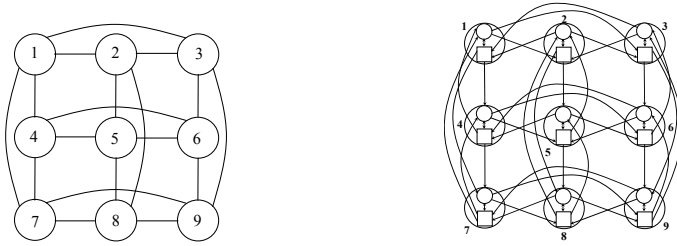


FIGURE 7. A two-dimensional ternary hypercube as a graph (left) and a digraph (right)

2. Hamilton Loops in Non-blocking System Area Networks

In section 2 we introduce non-blocking SANs as switches with the highest speed and complexity. Figures 7–9 demonstrate principles for building non-blocking self-routing distributed switches. First (Figure 7, left), consider a two-dimensional m -ary hypercube as a graph. If processors and switches are distinguished at its nodes, the hypercube becomes a digraph (Figure 7, right), which can be represented in bipartite form as a quasicomplete digraph (Figure 8). It can be built in the element base of $m \times m$ switches S_0 , $1 \times m$ demultiplexers, and $m \times 1$ multiplexers (Figure 9), thereby forming the distributed switch $S_1(N_1, m)$. The latter is a self-routing and non-blocking switch on arbitrary permutations for $N_1 = m^2$ users [4, 5].

Connections in the switch $S_1(9, 3)$ are given by different incident tables

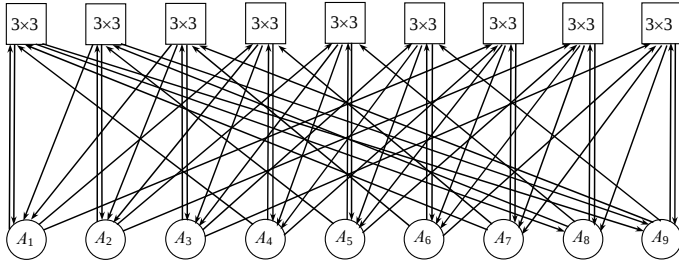
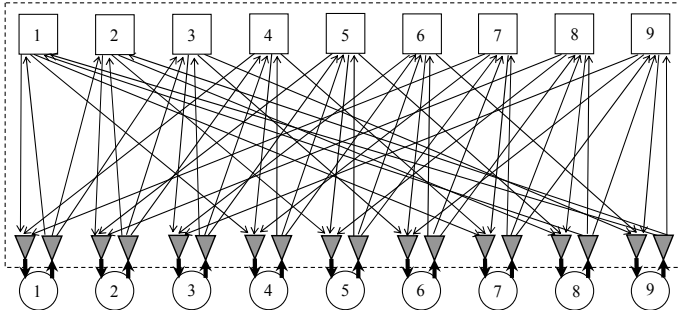

FIGURE 8. A quasicomplete digraph with $m = 3$

FIGURE 9. The switch $S_1(9, 3)$ with the quasicomplete digraph topology

TABLE 3. The incidence table for a quasicomplete digraph with $m = 3$

Switches	Arcs from users			Arcs to users		
	1	2	3	1	4	7
1	1	2	3	1	4	7
2	1	2	3	2	5	8
3	1	2	3	3	6	9
4	4	5	6	1	4	7
5	4	5	6	2	5	8
6	4	5	6	3	6	9
7	7	8	9	1	4	7
8	7	8	9	2	5	8
9	7	8	9	3	6	9

for arcs from users and to users (Table 3). They exist for any switches $S_1(N_1, m)$, see Table 4 and [4, 5].

TABLE 4. The incidence table for a quasicomplete digraph with arbitrary m

Switches	Channels from users				Channels to users			
1	1	2	...	m	1	$1 + m$	...	$1 + m(m - 1)$
2	1	2	...	m	2	$2 + m$	...	$2 + m(m - 1)$
...	1	2	...	m	...	...	...	...
m	1	2	...	m	m	$m + m$	...	$m + m(m - 1)$
$m + 1$	$m + 1$	$m + 2$	...	$m + m$	1	$1 + m$	...	$1 + m(m - 1)$
$m + 2$	$m + 1$	$m + 2$	...	$m + m$	2	$2 + m$	...	$2 + m(m - 1)$
...	$m + 1$	$m + 2$	...	$m + m$	...	...	...	...
$m + m$	$m + 1$	$m + 2$	...	$m + m$	m	$m + m$	...	$m + m(m - 1)$
...	...	...	...	...	...	...	...	...
$m^2 - m + 1$	$m_2 - m + 1$	$m_2 - m + 2$	...	m_2	1	$1 + m$	...	$1 + m(m - 1)$
...	$m^2 - m + 1$	$m_2 - m + 2$	...	m_2	2	$2 + m$	...	$2 + m(m - 1)$
$m^2 - 1$	$m_2 - m + 1$	$m_2 - m + 2$	...	m_2	...	...	...	...
m^2	$m_2 - m + 1$	$m_2 - m + 2$	...	m_2	m	$m + m$	...	$m + m(m - 1)$

TABLE 5. The incidence table for the non-blocking switch $S_2(27, 3)$

$K_1(9, 3)$	From users									To users								
1	1	2	3	10	11	12	19	20	21	1	4	7	10	13	16	19	22	25
2	1	2	3	10	11	12	19	20	21	2	5	8	11	14	17	20	23	26
3	1	2	3	10	11	12	19	20	21	3	6	9	12	15	18	21	24	27
4	4	5	6	13	14	15	22	23	24	1	4	7	10	13	16	19	22	25
5	4	5	6	13	14	15	22	23	24	2	5	8	11	14	17	20	23	26
6	4	5	6	13	14	15	22	23	24	3	6	9	12	15	18	21	24	27
7	7	8	9	16	17	18	25	26	27	1	4	7	10	13	16	19	22	25
8	7	8	9	16	17	18	25	26	27	2	5	8	11	14	17	20	23	26
9	7	8	9	16	17	18	25	26	27	3	6	9	12	15	18	21	24	27

Routing in the switch $S_1(N_1, m)$ is performed using the wormhole technique, i.e., by the output port numbers of the demultiplexer and switch.

With the internal scaling method [11], the non-blocking switch $S_{i+1}(N_{i+1}, m)$ with $N_{i+1} = m^{i+1}$ can be built from several non-blocking switches $S_i(N_i, m)$. More precisely, the switch $S_{i+1}(N_{i+1}, m)$ is obtained from N_1 switches $S_i(N_i, m)$ and a layer of N_{i+1} couples of $1 \times m$ demultiplexer and $m \times 1$ multiplexer. Each switch $S_i(N_i, m)$ is divided into parts of m ports; in each part, a separate switch $S_i(N_i, m)$ is constructed from all switches with initial numbering in each part and end-to-end numbering across all parts.

Table 5 provides an example of the incidence table of a switch $S_2(27, 3)$ from the incidence tables of switches $S_1(9, 3)$.

The connections in the switch $S_{i+1}(N_{i+1}, m)$ are constructed as follows. According to the source and sink numbers given, the switch $S_i(N_i, m)$ with both numbers (from the source and to the sink) is searched in the incidence table. The connection between the users is routed through the selected switch. For example, in Table 5, source 2 and sink 23 are connected through the switch with number 2.

Routing in the switch $S_i(N_i, m)$ is performed using the wormhole technique, i.e., by the output port numbers of the demultiplexers from the sources and further by the port numbers in switches $S_{i-1}(N_{i-1}, m)$.

The switch $S_1(N_1, m)$ has the complexity C_1 , expressed in the number of switching points; see formula (1). The number of channels in this switch, $L_1 = 2N_1^{\frac{3}{2}}$, is significantly smaller than in the complete graph.

$$(1) \quad \begin{aligned} C_1 &= 2(m^4 + m^3) = 2(N_1^2 + N_1^{\frac{3}{2}}) \\ L_1 &= 2m^3 = 2N_1^{\frac{3}{2}} \end{aligned}$$

The complexity and number of channels in the switch $S_i(N_i, m)$ are given by the recurrence relations $C_i = C_{i-1}N_1 + 2mN_i$ and $L_i = L_{i-1}N_1 + 2mN_i$. After straightforward transformations, we obtain the complexity and number of channels described by the formulas

$$(2) \quad \begin{aligned} S_i &= 2 \sum_{k=i+2}^{2(i+1)} m^k = 2 \sum_{k=i+2}^{2(i+1)} N_i^{\frac{k}{i+1}} \\ L_i &= 2 \sum_{k=i+2}^{2i+1} m^k = 2 \sum_{k=i+2}^{2i+1} N_i^{\frac{k}{i+1}}. \end{aligned}$$

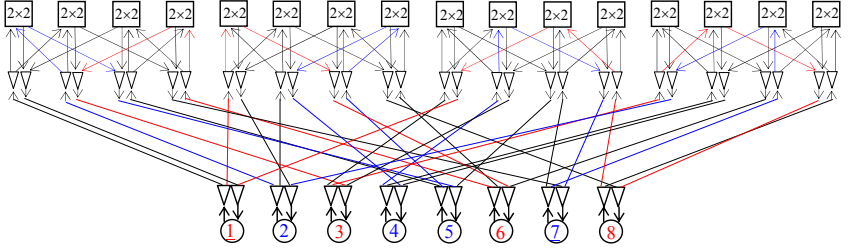
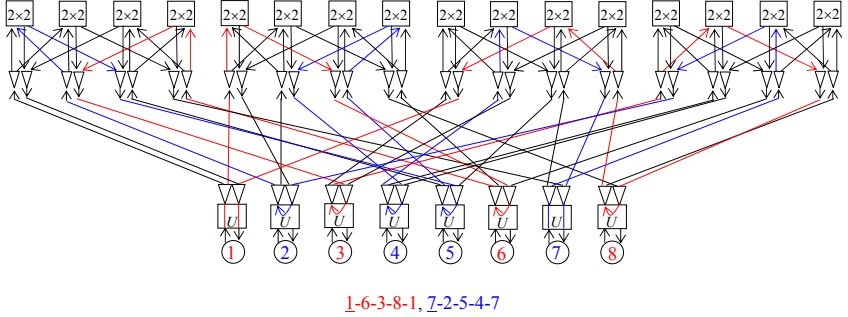
In non-blocking switches $S_i(N_i, m)$, it is possible to construct non-intersecting cyclic sequences of users (Hamilton graphs) with direct channels and completely defined nodes with initial data and the result. They are created by the sequential transmission of packets with intermediate sink addresses (in the form of packets of output port numbers) by the source nodes, with fixing the connections made.

As an illustrative example, Table 6 gives the incidence table of the switch $S_2(8, 2)$, and Figure 10 shows its circuit with two non-intersecting sequences of users. In this circuit, the connections for sequences 1-6-3-8-1 and 7-2-5-4-7 are highlighted in color, including initial users 1 and 7 creating them. The sequences are conflict-free by channels, and transmissions in them can be performed independently and in parallel.

Between each user and its channel splitters, it is possible to insert the executive unit U of the CCC operation from Section 1 (in the electronic version, see Figure 11). In the resulting SAN, different CCC operations can

TABLE 6. The incidence table for the non-blocking switch $S_2(8, 2)$ obtained from the switch $S_1(4, 2)$

$S_1(4, 2)$	From the source				To the sink			
1	1	2	5	6	3	5	7	10
2	1	2	5	6	4	6	8	20
3	4	7	8	3	7	5	3	30
4	3	4	7	8	2	8	6	40


 FIGURE 10. The switch $S_2(8, 2)$ with cyclic sequences 1-6-3-8-1 and 7-2-5-4-7

 FIGURE 11. The switch $S_2(8, 2)$ with executive nodes for CCC operations

be executed in cyclic sequences of nodes independently. The source nodes construct these cyclic sequences and set particular operations in them and, moreover, synchronize their execution (see section 1).

We compare the execution time of the group addition operation of numbers placed at N nodes of the networks similar to those in Figures 10 and 11.

In the first network, addition is executed by steps on a binary tree. The first step is to add, in parallel, the pairs of initial numbers at neighbor nodes, yielding $N/2$ intermediate sums of this step. At the i th step ($1 < i \leq \log_2 N$), the parallel addition of the intermediate sums of the $(i - 1)$ th step is executed.

Let us introduce the following notation: T_{num} is the time for transmitting a number to the network; τ is the average time for transmitting a number through the network to a neighbor node; δ is the time for executing addition at a node. Then the average addition time is given by $\sigma = (T_{\text{num}} + \delta) \log_2 N + \tau(1 + 2 + \dots + N/2) + \tau = (T_{\text{num}} + \delta) \log_2 N + \tau N$.

A Hamilton loop of nodes is formed in the time

$$\omega = N(\tau + T_{\text{add}}) + T_{\text{code}},$$

where T_{add} is the time for transmitting an address to the network and T_{code} is the time for transmitting the code of an operation under execution. Then the total time of the addition operation is

$$\Sigma = \sigma + \omega = (T_{\text{num}} + \delta) \log_2 N + 2\tau N + NT_{\text{add}} + T_{\text{code}}.$$

In the second network, the addition time (CCC operation) is given by

$$\theta = N\tau + NT_{\text{add}} + T_{\text{code}} + T_{\text{num}}.$$

As a matter of fact, it includes the formation time of a Hamilton loop of nodes. Indeed, the CCC operation is executed on the trailing edge of the transmission of the operation code immediately after the transmission of the address packet for forming the Hamilton loop; hence, the additional transmission time over the network τ is not required here.

Obviously, $\Sigma > 2\theta$. Note that the first operation is executed with program control of its steps whereas the second one is executed entirely in hardware, which even strengthens the above inequality.

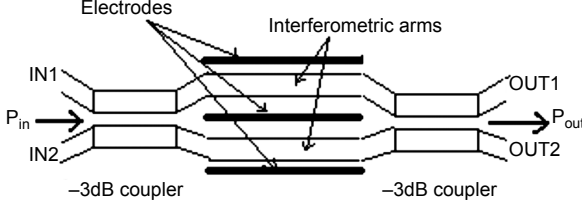


FIGURE 12. The switch $S_2(8, 2)$ with executive nodes for CCC operations

3. Non-blocking System Area Networks from Optoelectronic Switches Based on Mach-Zehnder Interferometers

An optical network for CCC can be built from optoelectronic switches based on Mach-Zehnder interferometers (MZI) [12, 13]. A conventional 2×2 MZI switch involving electrooptical properties (a passive EO-MZI structure) consists of two equal-length interferometric arms connected between two couplers; see Figure 12.

It operates as follows: the first coupler is used to uniformly split light into two parts, which experience a net phase change of $2\Delta\phi$ while passing through the interferometric arms. This phase difference is due to the pull-push effect of the field applied in opposite directions through the waveguides under the electrodes. As discovered for this structure, the output intensity is periodic with minima and maxima occurring at odd and even integer multiples of the voltage applied.

Therefore, light can be produced at the output ports through constructive or destructive interference phenomena. When the phase difference between the two arms is appropriately accumulated, it causes the appearance of recombined light in one of the output arms (i.e., the function of an optical 1×2 demultiplexer is realized).

Such switches can be used to assemble an optoelectronic 4×4 switch [13] with data transmission by optical signals and electronic control by a network node. Currently, the speed of such switches reaches up to 100 Gbps [14] and they have a fairly high packing density [15].

A non-blocking optical network can be based on tandem switching devices consisting of similar electronic and optoelectronic components. The former are involved in routing and route fixing as well as set the configuration of the latter. Tandem switching devices should have an identical configuration and operate in the master-slave mode for electronics and optics, respectively. The resulting non-blocking tandem network consists of copies of electronic and optoelectronic networks.

At the inputs/outputs of the second network, it is possible to place photonic executive nodes (Section 1) to execute CCC operations with higher speed and higher noise immunity.

Conclusion

In this paper, we have considered a method and circuitry for building non-blocking system area networks (SANs) to execute group operations in Hamilton loops of network nodes. Group operations are executed as common channel computing (CCC) operations with the highest speed. The SANs can have electronic and optoelectronic (tandem) implementation. The executive units of CCC operations can have both electronic and photonic implementation.

Note that the purely photonic implementation of executive units is one of the four innovations of this paper. The second innovation is the internal scaling method for non-blocking SANs with the digraph structure. The third one is a way to embed group operation circuits into a non-blocking SAN through the use of Hamilton loops. Finally, an innovation is the tandem method for organizing an optoelectronic non-blocking network.

References

- [1] S. M. Abramov, S. A. Stepanenko. “On approaches to developing software for a photonic computer”, Nacional’nyj Superkomp’yuternyj Forum (NSKF-2023) (Rossiya, Pereslavl’-Zalesskij, IPS imeni A.K. Ajlamazyana RAN, 28 noyabrya–01 dekabrya 2023 goda) (in Russian).  ⁴
- [2] I.V. Prangishvili, V.S. Podlazov, G.G. Stecyura. *Local microprocessor computer networks*, Glava shestaya, Nauka, M., 1984, 175 pp. (in Russian). ⁴

- [3] V.S. Podlazov. “Distributed arithmetic in a common channel based on photonic switches”, *Trudy 14-go Vserossijskogo soveshchaniya po problemam upravleniya (VSPU-2024)*, IPU RAN, M., 2024, ISBN 978-5-91450-276-5, pp. 2541–2546 (in Russian). ^{↑4}
- [4] V. S. Podlazov. “Multichannel non-blocking system area network with direct channels”, *Program Systems: Theory and Applications*, **13**:4 (55) (2022), pp. 47–76 (in Russian). ^{↑4, 12, 13}
- [5] V. S. Podlazov. “Multichannel non-blocking system area network with direct channels”, *Program Systems: Theory and Applications*, **14**:3 (58) (2023), pp. 115–138 (in Russian). ^{↑4, 12, 13}
- [6] K. Vytovtov, E. Barabanova, S. Zouhdi. “Optical switching cell based on metamaterials and ferrite films”, *2018 12th International Congress on Artificial Materials for Novel Wave Phenomena (Metamaterials)* (Espoo, Finland, 27 August 2018–01 September 2018), IEEE, 2018, ISBN 1-5386-4702-8, pp. 424–426. ^{↑6}
- [7] E. A. Barabanova, K. A. Vytovtov, T. T. Nguyen. “The control system elements of the new generation optical switching cell”, *Journal of Physics: Conference Series*, **1368**:2 (2019), id. 022002, 10 pp. ^{↑6}
- [8] M. R. Shcherbakov, L. Sheng, V. V. Zubyuk, A. Vaskin, P. P. Vabishchevich, G. Keeler, T. Pertsch, T. V. Dolgova, I. Staude, I. Brener, A. A. Fedyanin. “Ultrafast all-optical tuning of direct-gap semiconductor metasurfaces”, *Nature Communications*, **8**, id. 17, 6 pp. ^{↑7}
- [9] D. A. Belikov, E. V. Kaminskaya. *Computer Science. Fundamentals of Propositional Algebra*, Uchebno-metodicheskij kompleks, TGU, Tomsk, 2011 (in Russian). ^{↑9}
- [10] G. Roelkens, O. Raz, W. M. J. Green, S. Assefa, M. Tassaert, S. Keyvaninia, K. Vandoorne, D. Van Thourhout, R. Baets, Y. Vlasov. “Towards a low-power nanophotonic semiconductor amplifier heterogeneously integrated with SOI waveguides”, *7th IEEE International Conference on Group IV Photonics* (Beijing, China, 01–03 September 2010), IEEE, 2010, ISBN 9781424463466, pp. 16–18. ^{↑11}
- [11] M. F. Karavaj, V. S. Podlazov. “An invariant extension method for system area networks of multicore computational systems. An ideal system network”, *Autom. Remote Control*, 2010, no. 12, pp. 166–177. ^{↑15}
- [12] A. I. Stanley, G. Singh, J. Eke, H. Tsuda. “Mach–Zehnder interferometer: A review of a perfect all optical switching structure”, *Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing*, Springer, New Delhi, 2015, ISBN 978-81-322-2638-3, pp. 415–426. ^{↑19}

- [13] G. Singh, R. P. Yadav, V. Janyani. “Modeling of a high performance Mach-Zehnder interferometer all optical switch”, *Optica Applicata*, **42**:3 (2012), pp. 613–625.  [↑¹⁹](#)
- [14] W. M. J. Green, M. Yang, S. Assefa, J. Van Campenhout, B. G. Lee, C. V. Jahnke, F. E. Doany, C. L. Schow, J. A. Kash, Y. A. Vlasov. “Silicon electro-optic 4×4 non-blocking switch array for on-chip photonic networks”, *Proceedings Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011* (Los Angeles, California, United States, 6–10 March 2011), Optica Publishing Group, 2011, ISBN 978-1-55752-906-0, id. OThM1.  [↑](#)
- [15] T.-H. Yen, Y.-J. Hung. “Fabrication-insensitive CWDM (de)multiplexer based on cascaded Mach-Zehnder interferometers on silicon-on-insulator”, *Journal of Lightwave Technology*, **39**:1 (2020), pp. 146–153.  [↑](#)
- [16] Y. Gui, B. M. Nouri, M. Miscuglio, R. Amin, H. Wang, J. B. Khurgin, H. Dalir, V. J. Sorger. “100 GHz micrometer-compact broadband monolithic ITO Mach-Zehnder interferometer modulator enabling 3500 times higher packing density”, *Nanophotonics*, **11**:17 (2022), pp. 4001–4009.  [↑](#)

Received	24.03.2025;
approved after reviewing	17.04.2025;
accepted for publication	17.04.2025;
published online	24.09.2025.

Information about the author:



Viktor Sergeevich Podlazov

Doctor of Technical (Engineering) Science, assistant professor. Research interests: interconnect architectures and routing in supercomputer systems

 0000-0002-9175-1138
 e-mail: podlazov@ipu.ru
podlazov@gmail.com

The author declare no conflicts of interests.

УДК 004.724.2 + 004.272.43

 10.25209/2079-3316-2025-16-5-3-42


Групповые вычисления в неблокируемых системных сетях

 Виктор Сергеевич **Подлазов** 

Институт проблем управления имени В. А. Трапезникова РАН, Москва, Россия

 podlazov@ipu.ru

Аннотация. Рассмотрены неблокируемые оптоэлектронные сети, в которых выполняются групповые операции над множеством чисел за время передачи одного числа. Рассмотрено выполнение групповых операций исполнительными фотонными блоками, размещенными при каждом абоненте сети, и представлена схематехника этих блоков. Исследована возможность выполнения групповых операции в циклических последовательностях узлов, задаваемых гамильтоновыми графами. Предложены процедуры построения этих циклических последовательностей и выполнения в них групповых операций. Рассмотрена возможность построения неблокируемой электронной системной сети на любое число абонентов. Предложен способ построения оптоэлектронной сети как тандемной сети, в которой оптоэлектронные переключатели управляются их аналогами из электронной сети. *(Связанные тексты статьи на английском и на русском языках)*

Ключевые слова и фразы: неблокируемые системные сети, групповые операции, фотонные и оптоэлектронные переключатели и коммутаторы, гамильтоновы циклы узлов сети, тандемные неблокируемые сети

Для цитирования: Подлазов В. С. *Групповые вычисления в неблокируемых системных сетях* // Программные системы: теория и приложения. 2025. Т. 16. № 5(68). С. 3–42. (Англ.+русс.) https://psta.psisras.ru/read/psta2025_5_3-42.pdf

Введение

В настоящее время развиваются вычислительные системы на основе системных сетей, в которых данные передаются вместе с командами, образуя spatial data flow структуры [1]. Их частным случаем являются системы с параллельной обработкой данных в процессе их передачи по сети, в которых выполняется групповая операция над множеством чисел во время передачи по общему каналу сети одного числа, являющегося результатом операции. Такие операции называются «Вычисления в общем канале» (ВОК) [2] и они могут выполняться в прямых каналах без буферизации промежуточных значений («на лету»). Простейшими примерами операций ВОК являются суммирование множества чисел из разных узлов или нахождение из них максимального числа.

Канал является общим, если абоненты могут параллельно вести передачу сигналов в него с их наложением, которое можно использовать для групповых вычислений. Общий канал является прямым, если в нем имеются только задержки распространения сигналов или их комбинационные задержки и отсутствуют какие-либо тактовые задержки. Операции ВОК можно выполнять в циклических последовательностях сетевых узлов, образующих гамильтоновы графы. При этом операции ВОК на непересекающихся графах могут выполняться параллельно.

При каждом узле сети необходимо иметь исполнительные блоки для разных операций ВОК. Их можно строить в элементной базе разветвителей оптических каналов в чисто фотонной реализации (без электронного управления блоками) [3]. Здесь предполагается, во-первых, что сетевые узлы имеют возможность передавать/принимать данные в/из исполнительных блоков оптическими сигналами, во-вторых, что исполнительные блоки работают только с этими оптическими сигналами. Исполнительные блоки позволяют выполнять арифметические и логические операции над числами, представленными оптическими сигналами, без использования специальных фотонных АЛУ.

Циклические последовательности узлов с прямыми каналами можно строить в неблокируемых системных сетях, которые являются бесконфликтными на произвольных перестановках пакетов между узлами. В элементной базе коммутаторов и разветвителей каналов (мультиплексоров/демультиплексоров) можно строить неблокируемые сети с прямыми каналами любого размера по числу абонентов сети [4, 5].

На основе вышеперечисленных свойств в работе ставится задача разработать методику построения неблокируемой системной сети, которая обладает возможностью выполнения операций ВОК «на лету» на произвольном множестве узлов при фотонной реализации исполнительных блоков.

Для выполнения поставленной задачи потребовалось решить следующие частные задачи:

- (1) Выбор метода изменения значений разрядов в канале на основе их парафазного представления.
- (2) Выбор элементной базы для фотонных исполнительных блоков.
- (3) Разработка базовых элементов для фотонной реализации произвольных логических функций.
- (4) Выполнение синхронизации поразрядного наложения чисел в общем канале.
- (5) Разработка фотонных исполнительных блоков, которые выполняют операции ВОК с числами из разных сетевых узлов.
- (6) Разработка метода построения неблокируемой системной сети произвольного размера в элементной базе коммутаторов и разветвителей каналов заданного размера.
- (7) Разработка метода выполнения групповых операций в системной сети посредством формирования в ней гамильтоновых циклов сетевых узлов.
- (8) Разработка метода построения неблокируемой фотонной сети как части тандемной сети, в которой фотонные узлы находятся под управлением электронных узлов.

В разделе 1 рассматривается методика построения фотонных исполнительных блоков ВОК и предлагается схемотехника их реализации. В разделе 2 рассматривается методика построения неблокируемых системных сетей с прямыми каналами и образования в них гамильтоновых графов со способом встраивания в них исполнительных узлов ВОК. В разделе 3 рассматривается методика построений неблокируемых оптоэлектронных сетей на основе интерферометров Маха-Зандера, как базы для применения фотонных исполнительных блоков.

1. Распределенная арифметика на основе фотонных коммутаторов

Для выполнения операций ВОК логические переменные передаются по каналу в парафазном виде, т.е. по двум линиям 0 и 1. Значение 0 передается сигналом в линии 0 при отсутствии сигнала в линии 1, а значение 1 – сигналом в линии 1 при отсутствии сигнала в линии 0. Линии 0 и 1 проходят через конвертор-коммутатор C_i каждого i -го узла сети. Он выполняет унарную операцию O_i преобразования входной логической переменной x_i в выходную логическую переменную $y_i = O_i x_i$.

Операция O_i состоит из следующих преобразований: повторения $y_i = x_i$ («=»), отрицания $y_i = \neg x_i$ («¬»), перевода в значение $y_i = 0$ («0») и перевода в значение $y_i = 1$ («1»).

Конвертор используется в четырёх фиксированных состояниях, для которых операция O_i считается выполненной. Каждое из них задается некоторой конфигурацией пары отрезков волоконных линий связи, задаваемых значением парафазной управляющей переменной u_i .

На рисунке 1 представлена схемы использования таких пар конверторов для реализации некоторых логических функций на фотонных демульти-

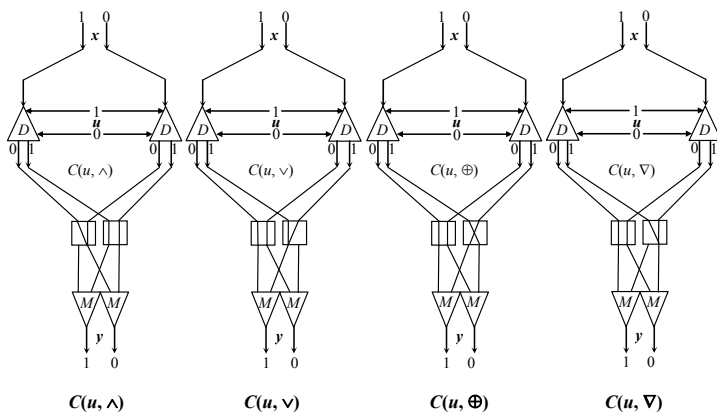


РИСУНОК 1. Схемная реализация логических функций из таблицы 1 для парафазных переменных

плексорах (D) и мультиплексорах (M) для парафазных переменных x , u и y .

Для этих схем в дальнейшем используются обозначения, приведенные на рисунке 1 снизу.

ТАБЛИЦА 1. Логические функции по состояниям конвертора для сумматора

Функция «И»				Функция «ИЛИ»				«Сложение по mod2»				«Начальный запуск»			
$\mathfrak{N} = \wedge$				$\mathfrak{N} = \vee$				$\mathfrak{N} = \oplus$				$\mathfrak{N} = \nabla$			
x	u	O	y	x	u	O	y	x	u	O	y	x	u	O	y
0	0	«0»	0	0	0	«=»	0	0	0	«=»	0	0	0	«=»	0
0	1	«=»	0	0	1	«1»	1	0	1	«¬»	1	0	1	«0»	0
1	0	«0»	0	1	0	«=»	1	1	0	«=»	1	1	0	«=»	1
1	1	«=»	1	1	1	«1»	1	1	1	«¬»	0	1	1	«0»	0

На рисунке 1 используются фотонные 1×2 демультиплексор D и 2×1 мультиплексор M , собираемые из нескольких слоев пленок метаматериалов с электрооптическими или магнитооптическими свойствами [6, 7]. Особенность их применения состоит в том, что для управления коммутацией используется оптический сигнал. В демультиплексоре они преобразуются в два внутренних электрических или магнитных

управляющих сигнала, которые подаются на управляющий слой, который задает два направления распространения информационного сигнала, поступившего на его оптический вход. В результате входной оптический сигнал направляется на один из двух оптических выходов.

В работе используется одна управляющая парафазная переменная, поэтому можно использовать другие однослойные устройства на два выхода (демультиплексор) или входа (мультиплексор), например, плёнки арсенида галлия [8]. Они позволяют строить фотонные логические элементы, управляемые светом напрямую.

В таблице 1 приводятся таблицы истинности логических операций, используемых в схемах на рисунке 1. Видно, что каждая операция $\&$ задается только парой состояний конвертора C_i , которые выбираются по значению переменной u_i . Как результат, на паре конверторов можно выполнить бинарную операцию $y_i = x_i \& u_i$ – любую из 16 логических операций.

На рисунке 2 представлена также схемная реализация для парафазных переменных мультиплексора (M_u) и демультиплексора (D_u), управляемых переменной u .

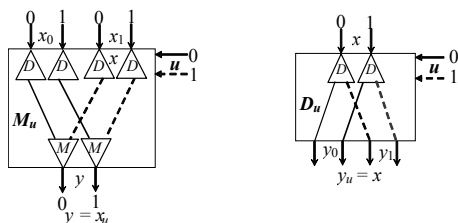


Рисунок 2. Схемная реализация мультиплексора и демультиплексора для парафазных переменных

При последовательном размещении узлов можно за время передачи по общему каналу одного числа образовать в нем сумму размещенных по узлам чисел или выделить из них максимальное число. Для этого числа передаются как двоичные числа последовательно по разрядам. Для суммирования они передаются младшими разрядами вперед, а для нахождения максимума – старшими разрядами.

В неблокируемых системных сетях всегда существуют циклические последовательности узлов в виде гамильтоновых графов, в каждом из которых существует путь, проходящий через каждый узел ровно один раз. Это свойство является прямым следствием свойств неблокируемых сетей, обеспечивающих бесконфликтную маршрутизацию на произвольных перестановках пакетов данных. Использование гамильтоновых циклов позволяет гарантировать бесконфликтную передачу данных и вложить в них линейные последовательности фотонных блоков, реализующих групповые операции ВОК. При этом удастся задать все источники и приемники данных.

В каждом цикле узлов выделяется исходный первый узел, с которого начинается их построение (см. следующий раздел), и который выдает команду выполняемой операции. Эта команда последовательно синхронизирует для всех узлов начало выполнения распределенной операции r . При каждом узле имеется набор исполнительных блоков $U(r)$, настроенных на выполнение отдельных операций. Команда операции выбирает конкретный исполнительный блок. Каждый исполнительный блок состоит из схем, выполняющих необходимые логические операции в определенной последовательности, заданной их межсоединениями.

На рисунке 3 представлены блоки $U_2(1)$ и $U_N(1)$ при втором и N -м узлах циклической последовательности, выполняющих операцию суммирования N чисел.

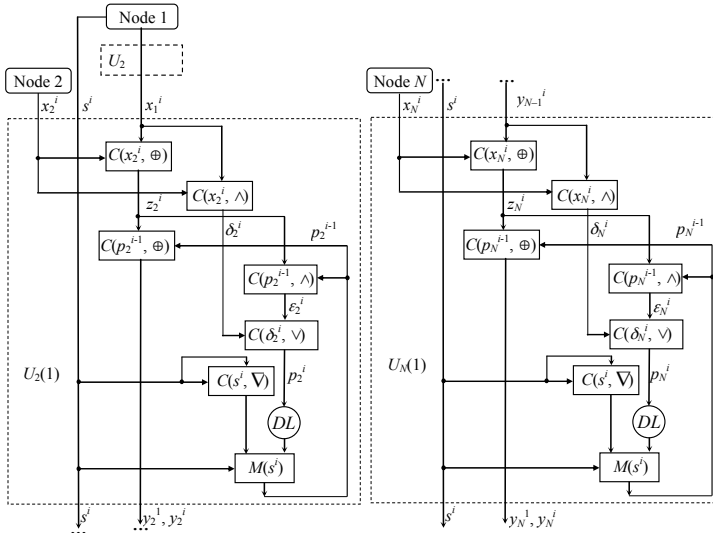


Рисунок 3. N-узеловой распределенный сумматор

Исходный первый узел сети не использует свой блок $U_1(1)$. Он выдает по парафазной линии значений слот X_1 , содержащий двоичное многоразрядное число, начиная с младших разрядов, а по парафазной синхролинии – слот S той же разрядности. В слоте X_1 разряды передаются переменными x_1^i , а в слоте S – переменными s^i , где i – это номер разряда. Первый разряд слота S имеет значение 1 ($s^1 = 1$), а остальные его разряды – значения 0 ($s^i = 0$ при $i > 1$). Точки на линиях обозначают узлы разделения световых сигналов посредством ветвления оптических линий. Блок $U_2(1)$ распределенного сумматора подсоединяется к первому и второму сетевым узлам.

Схемы на рисунке 3 получены на основе известной схемы одно-разрядного сумматора [9], к которой добавлена схема формирования нулевого значения переноса из предыдущего разряда для первого разряда и схема переноса значения в следующий разряд через линию задержки DL длительности в один такт.

На входы $U_2(1)$ подаются синхронно по разрядам слот S и слоты X_1 и X_2 , а на выходе образуется слот Y_2 , содержащий сумму чисел из слотов X_1 и X_2 . В этом блоке на конвертере $C(x_2^i, \oplus)$ складываются по модулю 2 значения переменных x_1^i и x_2^i , образуя промежуточный результат в виде переменной z_2^i . Последняя переменная вместе с переносом из предыдущего разряда p_2^{i-1} на конвертере $C(p_2^{i-1}, \oplus)$ образует выходную переменную y_2^i данного разряда (такта).

На конвертере $C(x_2^i, \wedge)$ из переменных x_1^i и x_2^i образуется промежуточная переменная δ_2^i , которая дальше используется на конвертере $C(\delta_2^i, \wedge)$ для образования значения переноса p_2^i в следующий разряд. Последний разряд, в свою очередь, образуется из промежуточной переменной ε_2^i , полученной на конвертере $C(p_2^{i-1}, \wedge)$ из результата z_2^i и переноса из предыдущего разряда (такта) p_2^{i-1} .

Схема $C(s^i, \nabla)$ образует значение переноса из предыдущего такта для первого разряда (такта). Через мультиплексор $M(s^i)$ пропускаются значение переноса в первом и других тактах.

Любой k -й узел сети ($2 < k \leq N$) передает в свой блок $U_k(1)$ слот X_k , содержащий значения переменных x_k^i для каждого разряда, на другой вход которого поступает слот Y_{k-1} , содержащий выходные разряды y_{k-1}^i из блока $U_{k-1}(1)$. Кроме того, в этот блок поступает слот S , прошедший через исполнительный блок предыдущего узла.

На рисунке 4 представлена временная диаграмма работы блока $U_2(1)$ в первых двух разрядах с использованием следующих обозначений. Период

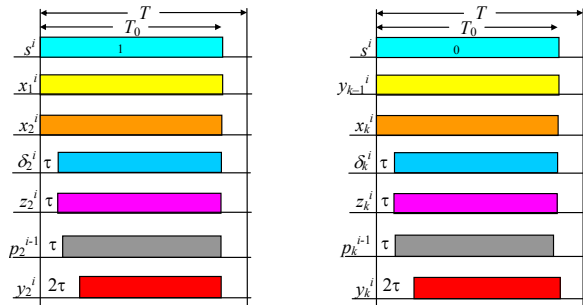


Рисунок 4. Временная диаграмма распределенного сумматора

разряда задается как T , длительность активных сигналов в нем – как T_0 , а время настройки конвертора или мультиплексора по управляющей

переменной u – как τ . Видно, что выходная переменная y_2^i во всех разрядах появляется с задержкой на 2τ и задается сигналами длительности $T_0 - 2\tau$.

На рисунке 4 (слева и справа) представлена временная диаграмма работы блоков $U_2(1)$ и $U_k(1)$ распределенного сумматора и в тех же обозначениях. Следует отметить, что правая диаграмма справедлива для любого блока $U_k(1)$ при $k > 2$. Это означает, что выходная переменная y_k^i появляется с задержкой на 2τ относительно начала синхросигналов s^i при любых k и i , т.е. при любом числе узлов и во всех разрядах чисел и задается сигналами длительности $T_0 - 2\tau$.

Наряду с рассмотренной выше операцией ВОК «суммирование» первый узел может задавать и операцию «максимизация», по которой находится наибольшее число из чисел, размещенных в узлах сети. По этой команде от исходного узла при каждом следующем узле выбирается исполнительный блок $U_k(2)$ – «максимизатор», выполняющий эту операцию, схема которого представлена на рисунке 5.

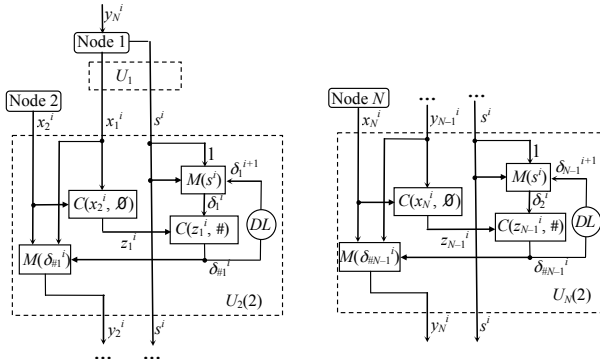


Рисунок 5. N -узовой распределенный «максимизатор»

В блоках $U_k(2)$ дополнительно используются схемы, которые выполняют логические функции, представленные в таблице 2.

ТАБЛИЦА 2. Логические функции по состояниям конвертора для «максимизатора»

Логическая функция «Переключение» для блока $U_k(2) (k \geq 2)$				Логическая функция «Повторение» для блока $U_k(2) (k \geq 2)$			
$\aleph = \emptyset$				$\aleph = \# = \wedge$			
x_{k-1}	x_k	O	z_{k-1}	z_{k-1}	δ_{k-1}	O	$\delta_{\#(k-1)}$
0	0	«1»	1	0	0	«0»	0
0	1	«=»	0	0	1	«=»	0
1	0	«1»	1	1	0	«0»	0
1	1	«=»	1	1	1	«=»	1

Первый узел сети выдает по парафазной линии значений в блок $U_2(2)$ слот X_1 , содержащий двоичное многоразрядное число, начиная

со старших разрядов, а по парафазной синхролинии – слот S той же разрядности. В слоте X_1 разряды передается переменными x_1^i , а в слоте S – переменными s^i , где i – это номер разряда. Первый разряд слота S имеет значение 1 ($s^1 = 1$), а остальные его разряды – значения 0 ($s^i = 0$ при $i > 1$).

Любой другой узел сети $1 < k \leq N$ передает по линии значений в свой блок $U_k(2)$ слот X_k , содержащий значения переменных x_k^i для каждого разряда. Кроме того, в этот блок поступает слот S , прошедший через исполнительный блок предыдущего узла.

На рисунке 5 (слева) представлена схема блока $U_2(2)$, подсоединяемого к первому и второму сетевым узлам распределенного «максимизатора». Точки на линиях обозначают узлы разделения световых сигналов посредством ветвления оптических линий. Проблема ослабления оптических сигналов может быть решена посредством использования фотонных усилителей на выходе блоков $U_k(2)$ [10].

Переменная δ_1^i должна находиться в значении $\delta_1^i = 1$ для всех разрядов $i < j$, для которых $x_1^i \geq x_2^i$, и находиться в значении $\delta_1^i = 0$ для всех разрядов $i > j$ после разряда j , для которого $x_1^j < x_2^j$. Такую последовательность значений δ_1 формируют конверторы $C(x_2^i, \emptyset)$ и $C(z_1^i, \#)$ совместно с мультиплексором $M(s^i)$ и линией задержки DL на один разряд.

Первый конвертор по значениям переменных x_1^i и x_2^i формирует значение промежуточной переменной z_1^i , реализуя логическую функцию «Переключение» по таблице 2 при $k \geq 2$. Второй конвертор формирует значение переменной $\delta_{\#1}^i$, реализуя логическую функцию «Повторение» по таблице 2 при $k \geq 2$. Мультиплексор $M(s^i)$ и линия задержки DL образуют значения переменной $\delta_1^{i+1} = \delta_{\#1}^i$ для следующего разряда.

Для тех разрядов $j < j$, для которых сохраняется значение $\delta_1^i = 1$, через мультиплексор $M(\delta_{\#1}^i)$ формируется значение выходной переменной $y_2^i = x_1^i$. Однако, для тех разрядов $j > j$, для которых сохраняется значение $\delta_1^i = 0$, через мультиплексор $M(\delta_{\#1}^i)$ формируется значение выходной переменной $y_2^i = x_2^i$. При этом для разряда j формируется значение $y_2^j = x_2^j = 1$.

На рисунке 5 (справа) приводится схема блока, $U_k(2)$, которая при $k > 1$ повторяет схему блока $U_2(2)$ с заменой x_1 на y_{k-1} и индекса 2 на индекс k .

На рисунке 6 приводится временная диаграмма формирования разных переменных для разрядов j и $j + 1$, где τ обозначает время настройки конверторов и мультиплексора, а T и T_0 – длительности разряда и активного сигнала в нем.

Видно, что активный сигнал укорачивается на время не большее 3τ ,

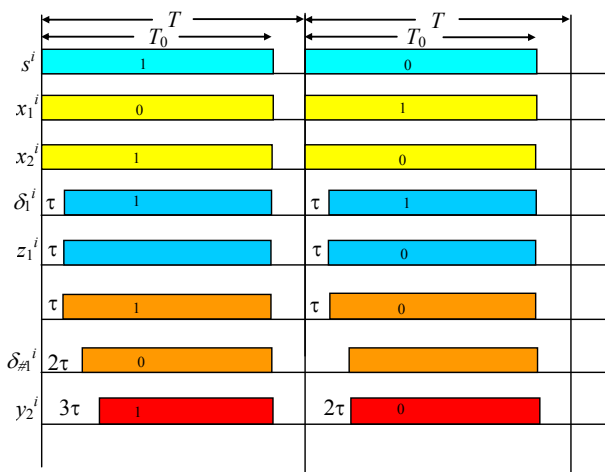


Рисунок 6. Временная диаграмма двухузлового «максимизатора» для j -го разряда

которое не зависит от значения j . В заключение раздела 1 необходимо отметить, что распределенные фотонные сумматор и «максимизатор» могут быть легко выполнены и в электронной элементной базе с заменой линии задержки DL на одноразрядный сдвиговый регистр.

2. Гамильтоновы циклы в неблокируемых системных сетях

В разделе 2 представляются неблокируемые системные сети как коммутаторы с наибольшим быстродействием и большой сложности.

На рисунках 7–9 представлены основания для построения неблокируемых самомаршрутизируемых распределенных коммутаторов.

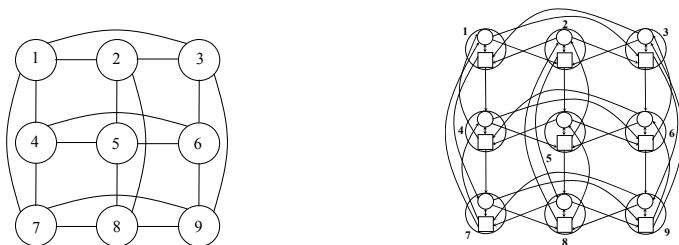


Рисунок 7. двумерный троичный гиперкуб при $m = 3$ как граф (слева) и оргграф (справа)

Сначала (рисунок 7 слева) это двумерный m -ичный гиперкуб как граф. Если в его узлах различать процессоры и коммутаторы, то он имеет форму оргграфа (рисунок 7 справа), который может быть представлен в двудольной форме как квазиполный оргграф (рисунок 8). Он может строиться в

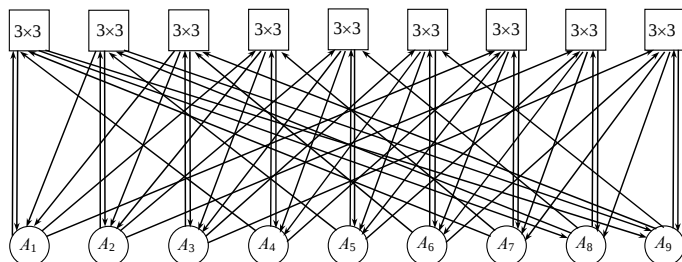


Рисунок 8. Квазиполный орграф при $m = 3$

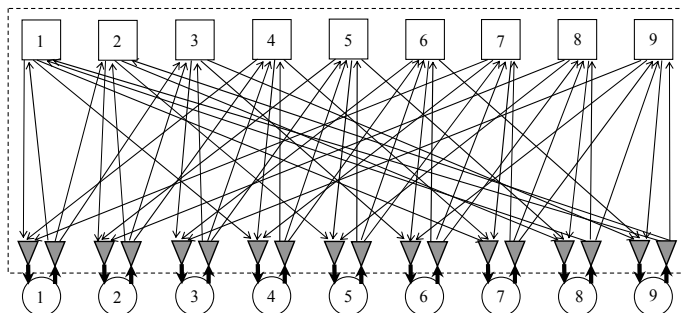


Рисунок 9. Коммутатор $K_1(9, 3)$ с топологией квазиполного орграфа

элементной базе из $m \times m$ -коммутаторов K_0 , $1 \times m$ -демультиплексоров и $m \times 1$ -мультиплексоров (рисунок 9) и образовать распределенный коммутатор $K_1(N_1, m)$, который является самомаршрутизируемым и неблокируемым коммутатором на произвольных перестановках для $N_1 = m^2$ абонентов [4, 5].

Соединения в коммутаторе $K_1(9, 3)$ задаются разными таблицами инцидентности для дуг от абонентов и дуг к абонентам (таблица 3). Они

Таблица 3. Таблица инцидентности для квазиполного орграфа при $m = 3$

Коммутаторы	Дуги от абонентов			Дуги к абонентам		
1	1	2	3	1	4	7
2	1	2	3	2	5	8
3	1	2	3	3	6	9
4	4	5	6	1	4	7
5	4	5	6	2	5	8
6	4	5	6	3	6	9
7	7	8	9	1	4	7
8	7	8	9	2	5	8
9	7	8	9	3	6	9

существуют для любых коммутаторов $K_1(N_1, m)$ (таблица 4) [4, 5].

Таблица 4. Таблица инцидентности для квазиполного орграфа при произвольном m

Коммутаторы	Каналы от абонентов				Каналы к абонентам			
1	1	2	...	m	1	$1+m$	...	$1+m(m-1)$
2	1	2	...	m	2	$2+m$	...	$2+m(m-1)$
...	1	2	...	m	...	...	...	...
m	1	2	...	m	m	$m+m$	...	$m+m(m-1)$
$m+1$	$m+1$	$m+2$	...	$m+m$	1	$1+m$	...	$1+m(m-1)$
$m+2$	$m+1$	$m+2$	...	$m+m$	2	$2+m$	...	$2+m(m-1)$
...	$m+1$	$m+2$	...	$m+m$	...	...	...	...
$m+m$	$m+1$	$m+2$	...	$m+m$	m	$m+m$	...	$m+m(m-1)$
...	...	...	...	...	...	...	...	...
m^2-m+1	m_2-m+1	m_2-m+2	...	m_2	1	$1+m$	...	$1+m(m-1)$
...	m^2-m+1	m_2-m+2	...	m_2	2	$2+m$	...	$2+m(m-1)$
m^2-1	m_2-m+1	m_2-m+2	...	m_2	...	...	...	...
m^2	m_2-m+1	m_2-m+2	...	m_2	m	$m+m$	...	$m+m(m-1)$

Маршрутизация в коммутаторе $K_1(N_1, m)$ осуществляется методом червоточкины – по номерам выходных портов демультиплексора и коммутатора.

Существует [11] метод внутреннего масштабирования, который позволяет строить из неблокируемых коммутаторов $K_i(N_i, m)$ неблокируемый коммутатор $K_{i+1}(N_i + 1, m)$ с $N_{i+1} = m^{i+1}$. Коммутатор $K_{i+1}(N_i + 1, m)$ строится из N_1 коммутаторов $K_i(N_i, m)$ и слоя из N_{i+1} демультиплексоров $1 \times m$ и мультиплексоров $m \times 1$. Каждый коммутатор $K_i(N_i, m)$ разбивается на доли по m портов, и в каждой доле по всем коммутаторам строится отдельный коммутатор $K_i(N_i, m)$ с исходной нумерацией в каждой доле и сквозной нумерацией по всем долям.

В таблице 5 приводится пример построения таблицы инцидентности коммутатора $K_2(27, 3)$ из таблиц инцидентности коммутаторов $K_1(9, 3)$.

Таблица 5. Таблица инцидентности для неблокируемого коммутатора $K_2(27, 3)$

$K_1(9, 3)$	От абонентов									К абонентам								
1	1	2	3	10	11	12	19	20	21	1	4	7	10	13	16	19	22	25
2	1	2	3	10	11	12	19	20	21	2	5	8	11	14	17	20	23	26
3	1	2	3	10	11	12	19	20	21	3	6	9	12	15	18	21	24	27
4	4	5	6	13	14	15	22	23	24	1	4	7	10	13	16	19	22	25
5	4	5	6	13	14	15	22	23	24	2	5	8	11	14	17	20	23	26
6	4	5	6	13	14	15	22	23	24	3	6	9	12	15	18	21	24	27
7	7	8	9	16	17	18	25	26	27	1	4	7	10	13	16	19	22	25
8	7	8	9	16	17	18	25	26	27	2	5	8	11	14	17	20	23	26
9	7	8	9	16	17	18	25	26	27	3	6	9	12	15	18	21	24	27

Построение соединений в коммутаторе $K_{i+1}(N_{i+1}, m)$ осуществляется следующим образом. По номерам источника и приемника в таблице инцидентности ищется коммутатор $K_i(N_i, m)$, в который входят оба номера: от источника и к приемнику. Соединение между абонентами прокладывается через выбранный коммутатор. Например, в таблице 5 источник 2 и приемник 23 соединяются через коммутатор с номером 2.

Маршрутизация в коммутаторе $K_i(N_i, m)$ осуществляется по-прежнему методом червоточины – по номерам выходных портов демультимплексов от источников и далее по номерам портов в коммутаторах $K_{i-1}(N_{i-1}, m)$.

Коммутатор $K_1(N_1, m)$ имеет сложность S_1 , выражаемую в числе точек коммутации по формуле (1). Число каналов в нем $L_1 = 2N_1^{\frac{3}{2}}$ существенно меньше, чем в полном графе:

$$(1) \quad \begin{aligned} S_1 &= 2(m^4 + m^3) = 2(N_1^2 + N_1^{\frac{3}{2}}) \\ L_1 &= 2m^3 = 2N_1^{\frac{3}{2}}. \end{aligned}$$

Сложность и число каналов коммутатора $K_i(N_i, m)$ задаются следующими рекуррентными соотношениями $S_i = S_{i-1}N_1 + 2mN_i$ и $L_i = L_{i-1}N_1 + 2mN_i$. В результате их разрешения получаем значения сложности и числа каналов, задаваемые формулами (2)

$$(2) \quad \begin{aligned} S_i &= 2 \sum_{k=i+2}^{2(i+1)} m^k = 2 \sum_{k=i+2}^{2(i+1)} N_i^{\frac{k}{i+1}} \\ L_i &= 2 \sum_{k=i+2}^{2i+1} m^k = 2 \sum_{k=i+2}^{2i+1} N_i^{\frac{k}{i+1}}. \end{aligned}$$

В неблокируемых коммутаторах $K_i(N_i, m)$ можно строить непересекающиеся циклические последовательности абонентов (гамильтоновы графы), связанных прямыми каналами, в которых полностью определены узлы с исходными данными и полученным результатом. Они создаются последовательной передачей исходными узлами пакетов адресов промежуточных приемников (в виде цугов номеров выходных портов) с фиксацией проложенных соединений.

Для примера в таблице 6 приводится таблица инцидентности коммутатора $K_2(8, 2)$, а на рисунке 10 приводится его схема с двумя непересекающимися последовательностями абонентов. На схеме цветом выделены соединения для последовательностей 1-6-3-8-1 и 7-2-5-4-7, с исходными абонентами 1 и 7, которые их и создают. Эти последовательности являются бесконфликтными по каналам, и передачи в них могут осуществляться независимо и параллельно.

Между каждым абонентом и его разветвителями каналов можно вставить исполнительный блок U операции ВОК из раздела 1 в электронном исполнении (рисунок 11). В результате получится системная сеть,

ТАБЛИЦА 6. Таблица инцидентности для неблокируемого коммутатора $K_2(8, 2)$, получаемого из коммутатора $K_1(4, 2)$

$K_1(4, 2)$	От источника				К приемнику			
1	1	2	5	6	3	5	7	10
2	1	2	5	6	4	6	8	20
3	4	7	8	3	7	5	3	30
4	3	4	7	8	2	8	6	40

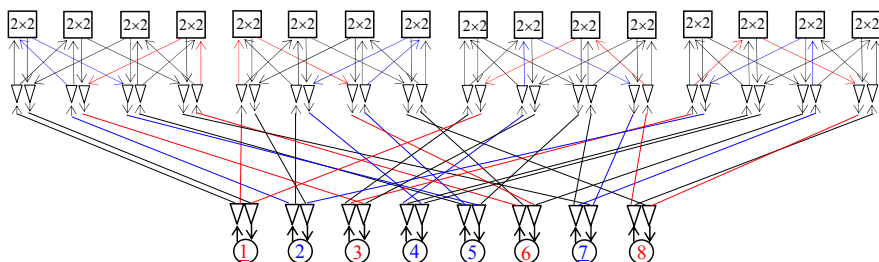


РИСУНОК 10. Коммутатор $K_2(8, 2)$ с циклическими последовательностями 1-6-3-8-1 и 7-2-5-4-7

в циклических последовательностях узлов которой можно независимо выполнять разные операции ВОК. При этом исходные узлы не только строят эти циклические последовательности, задают в них конкретные операции, но и синхронизируют их исполнение (см. раздел 1).

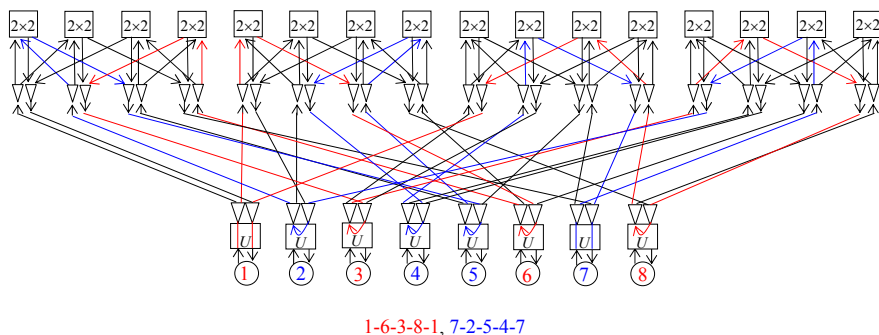


РИСУНОК 11. Коммутатор $K_2(8, 2)$ с исполнительными узлами для операций ВОК

Сравним время выполнения групповой операции суммирования чисел, размещенных в N узлах сетей, которые аналогичны сетям на рисунке 10 и рисунок 11.

В первой сети суммирование осуществляется шагами по двоичному дереву. На 1-ом шагу производится параллельное суммирование пар исходных чисел в соседних узлах с образованием $N/2$ промежуточных сумм 1-го шага. На i -ом шагу ($1 < i \leq \log_2 N$) производится параллельное суммирование промежуточных сумм $(i - 1)$ -го шага.

Обозначим время передачи числа в сеть как $T_{\text{ч}}$, среднее время передачи числа по сети в соседний узел как τ , а время суммирования в узле как δ . Тогда время суммирования в среднем задается выражением:

$$\sigma = (T_{\text{ч}} + \delta) \log_2 N + \tau(1 + 2 + \dots + N/2) + \tau = (T_{\text{ч}} + \delta) \log_2 N + \tau N.$$

Время образования гамильтонова цикла узлов задается как

$$\omega = N(\tau + T_{\text{а}}) + T_{\text{к}},$$

где $T_{\text{а}}$ это время передачи адреса в сеть, а $T_{\text{к}}$ – время передачи кода исполняемой операции. Тогда полное время операции суммирования выражается как

$$\Sigma = \sigma + \omega = (T_{\text{ч}} + \delta) \log_2 N + 2\tau N + NT_{\text{а}} + T_{\text{к}}.$$

Во второй сети время суммирования как операции ВОК задается выражением:

$$\theta = N\tau + NT_{\text{а}} + T_{\text{к}} + T_{\text{ч}}.$$

В нем уже учтено время формирования гамильтонова цикла узлов. Дело в том, что операция ВОК выполняется на заднем фронте передачи кода операции сразу после передачи адресного пакета формирования гамильтонова цикла, и не требует дополнительного времени передачи по сети τ .

Видно, что $\Sigma > 2\theta$. При этом нужно учитывать, что первая операция выполняется с программным управлением ее шагов, а вторая операция выполняется полностью аппаратно, что еще более усиливает приведенное неравенство.

3. Неблокируемые системные сети из оптоэлектронных переключателей на базе интерферометров Маха-Зандера

Оптическую сеть для ВОК можно построить из оптоэлектронных переключателей на базе интерферометров Маха-Зандера (MZI) [12, 13]. Обычный 2×2 -переключатель MZI на основе электрооптических свойств (пассивная структура EO-MZI) состоит из двух интерферометрических плеч равной длины, соединенных между двумя ответвителями, как показано на рисунке 12.

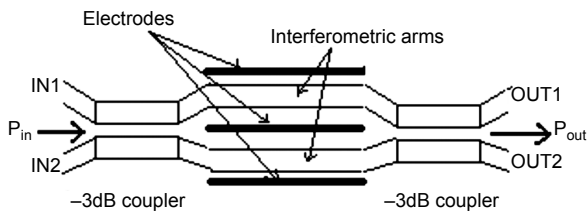


Рисунок 12. Общая схема переключателя EO-MZI

Его работу можно понять следующим образом: первый ответвитель используется для равномерного разделения света на две части, которые при прохождении через интерферометрические плечи испытывают чистое изменение фазы на $2\Delta\phi$. Эта разность фаз обусловлена эффектом «тяги-толкай», вызванным полем, приложенным в противоположных направлениях через волноводы под электродами.

Как было замечено, с этой структурой выходная интенсивность является периодической с минимумами и максимумами, возникающими при нечетных и четных целых кратных приложенного напряжения. Поэтому свет может быть получен на выходных портах посредством явлений конструктивной или деструктивной интерференции. При соответствующем накоплении разности фаз между двумя плечами приводит к появлению рекомбинированного света в одном из выходных плеч, т.е. реализуется функция оптического демультиплексора 1×2 .

Из таких переключателей можно собрать оптоэлектронный коммутатор 4×4 [13], в котором передача данных через коммутатор ведется оптическими сигналами, а управление коммутатором осуществляется узлом сети электронным образом. В настоящее время быстродействие таких переключателей доходит до 100 Гбит/сек [14] и они имеют достаточно высокую плотность упаковки [15].

Неблокируемую оптическую сеть можно создать на основе тандемных коммутирующих устройств, состоящих из аналогичных электронных и оптоэлектронных компонент. Первые участвуют в прокладке и фиксации маршрутов, и задают конфигурацию вторых. Тандемные коммутирующие устройства должны иметь идентичную конфигурацию и работать в режиме master (электроника)-slave (оптика).

Таким образом создается тандемная неблокируемая сеть, состоящая из копий электронной и оптоэлектронной сетей. На входах/выходах второй сети можно разместить фотонные исполнительные узлы (раздел 1) для выполнения операций ВОК с большим быстродействием и с большей помехозащищенностью.

Заключение

Рассмотрена методика и схемотехника построения неблокируемых системных сетей, обеспечивающих выполнение групповых операций в гамильтоновых циклах сетевых узлов. Групповые операции выполняются как операции вычислений в общем канале (ВОК) с наибольшим быстродействием. Системные сети могут иметь электронную и оптоэлектронную (тандемную) реализацию. Исполнительные блоки выполнения операций ВОК могут иметь как электронную, так и фотонную реализацию.

Необходимо отметить, что чисто фотонная реализация исполнительных блоков является одной из четырех новаций данной работы. Второй новацией является способ внутреннего масштабирования неблокируемых системных сетей с орграфовой структурой. Третьей новацией является способ встраивания схем выполнения групповых операций в неблокируемую системную сеть посредством использования гамильтоновых циклов. Наконец, новацией является тандемный способ организации оптоэлектронной неблокируемой сети.

Список использованных источников

- [1] Абрамов С. М., Степаненко С. А. *О подходах к разработке программного обеспечения для фотонной вычислительной машины*, Национальный Суперкомпьютерный Форум (НСКФ-2023) (Россия, Переславль-Залесский, ИПС имени А.К. Айламазяна РАН, 28 ноября–01 декабря 2023 года).  ²⁴
- [2] Прангишвили И.В., Подлазов В.С., Стецюра Г.Г. *Локальные микропроцессорные вычислительные сети*, Глава шестая.— М.: Наука.— 1984.— 175 с.

↑²⁴

- [3] Подлазов В.С. *Распределенная арифметика в общем канале на основе фотонных коммутаторов* // Труды 14-го Всероссийского совещания по проблемам управления (ВСПУ-2024), М.: ИПУ РАН.– 2024.– ISBN 978-5-91450-276-5.– С. 2541–2546. ✨ ↑²⁴
- [4] Подлазов В.С. *Самомаршрутизируемая неблокируемая системная сеть с прямыми каналами: сложность и быстродействие* // Программные системы: теория и приложения.– 2022.– Т. 13.– № 4 (55).– С. 47–76. doi URL ↑²⁴, 33
- [5] Подлазов В.С. *Разные неблокируемые самомаршрутизируемые системные сети с прямыми каналами* // Программные системы: теория и приложения.– 2023.– Т. 14.– № 3 (58).– С. 115–138. doi URL ↑²⁴, 33
- [6] Vytovtov K., Barabanova E., Zouhdi S. *Optical switching cell based on metamaterials and ferrite films* // 2018 12th International Congress on Artificial Materials for Novel Wave Phenomena (Metamaterials) (Espoo, Finland, 27 August 2018–01 September 2018).– IEEE.– 2018.– ISBN 1-5386-4702-8.– Pp. 424–426. doi ↑²⁶
- [7] Barabanova E. A., Vytovtov K. A., Nguyen T. T. *The control system elements of the new generation optical switching cell* // Journal of Physics: Conference Series.– 2019.– Vol. 1368.– No. 2.– id. 022002.– 10 pp. doi ↑²⁶
- [8] Shcherbakov M. R., Sheng L., Zubyuk V. V., Vaskin A., Vabishchevich P. P., Keeler G., Pertsch T., Dolgova T. V., Staude I., Brener I., Fedyanin A. A. *Ultrafast all-optical tuning of direct-gap semiconductor metasurfaces* // Nature Communications.– Vol. 8.– id. 17.– 6 pp. doi ↑²⁷
- [9] Беликов Д. А., Каминская Е. В. *Информатика. Основы алгебры высказываний, Учебно-методический комплекс.*– Томск: ТГУ.– 2011. URL ↑²⁹
- [10] Roelkens G., Raz O., Green W. M. J., Assefa S., Tassaert M., Keyvaninia S., Vandoorne K., Van Thourhout D., Baets R., Vlasov Y. *Towards a low-power nanophotonic semiconductor amplifier heterogeneously integrated with SOI waveguides* // 7th IEEE International Conference on Group IV Photonics (Beijing, China, 01–03 September 2010).– IEEE.– 2010.– ISBN 9781424463466.– Pp. 16–18. doi ↑³¹
- [11] Каравай М. Ф., Подлазов В.С. *Метод инвариантного расширения системных сетей многопроцессорных вычислительных систем. Идеальная системная сеть* // Автомат. и телемех.– 2010.– № 12.– С. 166–177. % ↑³⁵
- [12] Stanley A. I., Singh G., Eke J., Tsuda H. *Mach-Zehnder interferometer: A review of a perfect all optical switching structure* // Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing, New Delhi: Springer.– 2015.– ISBN 978-81-322-2638-3.– Pp. 415–426. doi ↑³⁹
- [13] Singh G., Yadav R. P., Janyani V. *Modeling of a high performance Mach-Zehnder interferometer all optical switch* // Optica Applicata.– 2012.– Vol. 42.– No. 3.– Pp. 613–625. doi ↑³⁹

- [14] Green W. M. J., Yang M., Assefa S., Van Campenhout J., Lee B. G., Jahnes C. V., Doany F. E., Schow C. L., Kash J. A., Vlasov Y. A. *Silicon electro-optic 4×4 non-blocking switch array for on-chip photonic networks* // *Proceedings Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2011* (Los Angeles, California, United States, 6–10 March 2011). – Optica Publishing Group. – 2011. – ISBN 978-1-55752-906-0. – id. OThM1.  ^{↑39}
- [15] Yen T.-H., Hung Y.-J. *Fabrication-insensitive CWDM (de)multiplexer based on cascaded Mach-Zehnder interferometers on silicon-on-insulator* // *Journal of Lightwave Technology*. – 2020. – Vol. **39**. – No. 1. – Pp. 146–153.  ^{↑39}
- [16] Gui Y., Nouri B. M., Miscuglio M., Amin R., Wang H., Khurgin J. B., Dalir H., Sorger V. J. *100 GHz micrometer-compact broadband monolithic ITO Mach-Zehnder interferometer modulator enabling 3500 times higher packing density* // *Nanophotonics*. – 2022. – Vol. **11**. – No. 17. – Pp. 4001–4009.  [↑]

Поступила в редакцию 24.03.2025;
одобрена после рецензирования 17.04.2025;
принята к публикации 17.04.2025;
опубликована онлайн 24.09.2025.

Рекомендовал к публикации

к.ф.-м.н. С. А. Романенко

Информация об авторе:



Виктор Сергеевич Подлазов

Д. т. н., гл. н. с. Института проблем управления им. В. А. Трапезникова РАН. Научные интересы: архитектуры интерконнекта и маршрутизация в суперкомпьютерных системах



0000-0002-9175-1138

e-mail: podlazov@ipu.ru
podlazov@gmail.com

Декларация об отсутствии личной заинтересованности: *благополучие автора не зависит от результатов исследования.*