

# Structure and Implementation Principles of a Photonic Computer

Sergey Stepanenko<sup>1,\*</sup>

<sup>1</sup>Federal State Unitary Enterprise, Russian Federal Nuclear Center, All-Russian Research Institute of Experimental Physics, FSUE RFNC-VNIIEF, RU- 607188, Sarov, Russia

**Abstract.** A structure and implementation principles of a photonic computer are proposed. Its operation is based on the effects of interaction between coherent light wave systems generated by a laser source. The performance of photonic computers, their power consumption, and physical dimensions are estimated. These estimates indicate a great potential advantage of photonic computers over electronic ones. In particular, analysis carried out in this study and the obtained estimates demonstrate that a photonic computer working on the light with a wavelength of 1530 can operate 104 – 105 times faster than up-to-date electronic computers with the same power consumption. Light waves of different wavelengths do not interact with each other. With a proper OLG implementation, several computations represented by light waves of different wavelengths can be run on a single photonic computer.

## 1 Introduction

A photonic computer processes information in the form of light carried by photons, hence the name. Its operation is based on the interaction between coherent light wave systems generated by a laser source [1].

Building a photonic computer has become relevant because electronic computers have encountered fundamental difficulties improving their performance, while the prospect of implementation and application of quantum computers is uncertain [2].

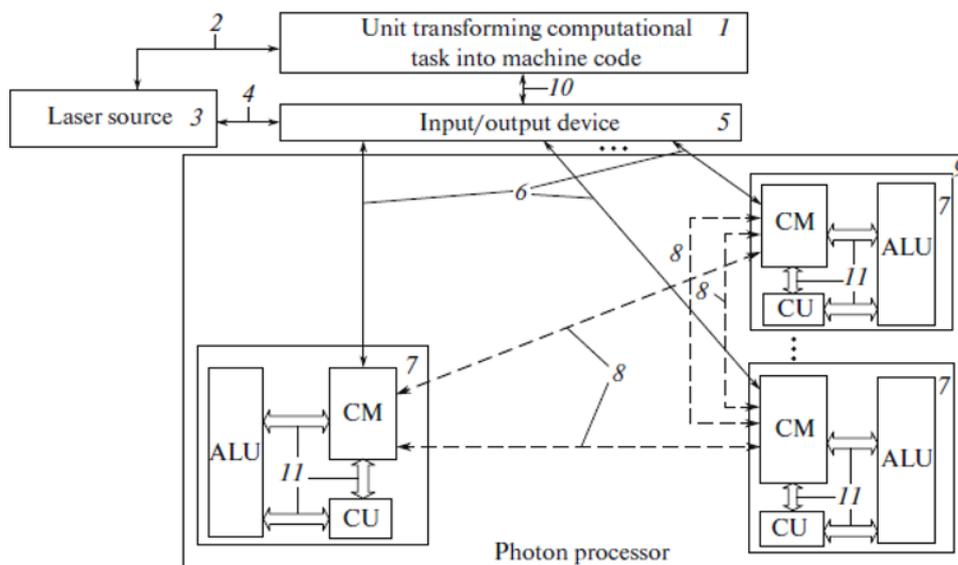
In contrast to its equivalents [3, 4], the photonic computer discussed in this paper is implemented using passive optical logic gates [5,6]. The classes of problems solved by photonic and electronic computers are the

same. This paper is an extended version of publications [7,8].

## 2 Operation algorithm and structure of a photonic computer

A program written in a high-level language is translated by an electronic computer into an electronic program of a photonic computer and then converted into an optical program executed by a photonic processor. When computations are completed, the output is converted by an interface into an electronic form and transmitted to the electronic computer.

The structure of a photonic computer is shown in



**Fig.1.** Structure of a photonic computer

\* e-mail: [ssa@vniief.ru](mailto:ssa@vniief.ru)

A computation is launched by a starting pulse sent by device 1 through channel 2 to laser source 3. Light generated by source 3 passes through optical channel 4 and arrives at input/output device 5, where it is split into beams, the number of which equals the number of bits delivered simultaneously through optical channels 6 to processor elements (PE) 7. The processor elements are connected by channels 8 into photonic processor 9. The beams are then converted in device 5 in accordance with the electronic representation of the input information delivered through electronic channel 10 from device 1 (which can be an electronic computer) to produce an optical form of the program transmitted through channels 6 to processor elements 7 for execution. The processor elements contain arithmetic/logic units (ALU), switches (SW), and control units (CU) connected by optical channels 11. A computation is a sequence of beam interactions in passive optical logic gates (OLG) [5, 6] that implement processor elements 7. The functions of the OLGs are identical to those of electronic logic gates [9]. This makes it possible to use architectural implementations of ALUs, SWs and CUs known from electronic computing for photonic computing; timing can be performed using optical delay lines.

The processor elements in the photonic processor are connected by channels 8 into a multiprocessor system of any known topology, such as a 3D torus, a  $\Gamma_n$  hypercube, etc. [10].

The total number of operations performed is determined by the power budget [11] or coherence length [12].

After the power budget is exhausted, the information is regenerated. Once the coherence length is reached, the information is sent back through channels 6 to device 5, where it is converted to the electronic form and then again to the coherent optical form transmitted to the processor elements.

### 3 Computing Principles

Optical information constantly moves in space. Preventing its delays and power losses requires the following:

- operations must be performed without memory calls as soon as operands are available according to the data flow control discipline [9];
- information must be processed by conflict-free routes [10] involving only free processor elements and channels at a certain known time step.

The expression to be computed is represented in the reverse Polish notation [9], which is used to construct a multilevel structure (MLS) of the algorithm [13]. The vertices of one MLS level correspond to instructions executed independently (in parallel) at the same time step  $t$ .

The algorithm's MLS yields a processor graph. Its vertices correspond to the processor elements performing operations at time steps  $t$ . They are connected by edges belonging to conflict-free routes of channels 8. The processor elements use these channels to exchange their identifiers, i.e. codes of operations, operands and synchronous idle characters, denoted by  $\emptyset$ . Operations

are assigned to the processor elements that are free of other tasks at the time step  $t$ ; these processor elements are identified in the course of preparing the program in device 1 depending on the number of processor elements, their communication topology, and membership in conflict-free routes.

The program is constructed based on the processor graph by replacing its vertices with photonic computer instructions.

During the course of a computation, each identifier is accompanied by a current value of  $t=0,1,\dots, m$ , where  $m$  is the algorithm's MLS depth [13]. The value of  $t$  increases by one with each run across the processor element.

Idle processor elements receive and transmit the symbol  $\emptyset$  and the values of  $t$  at each time step (increasing  $t$  by one).

The processor element is engaged only if identical successive values of  $t$  are received through all of its channels 8.

### 4 Example of a Computation

Let us draw a processor graph to calculate the value of the expression  $A=a+(b+c)\times d$  by a photonic computer, whose processor elements are connected to a  $\Gamma_3$  (three-dimensional hypercube) topology [10], as shown in Fig.2. The numbers in Fig.1 and Fig.2 are the same.

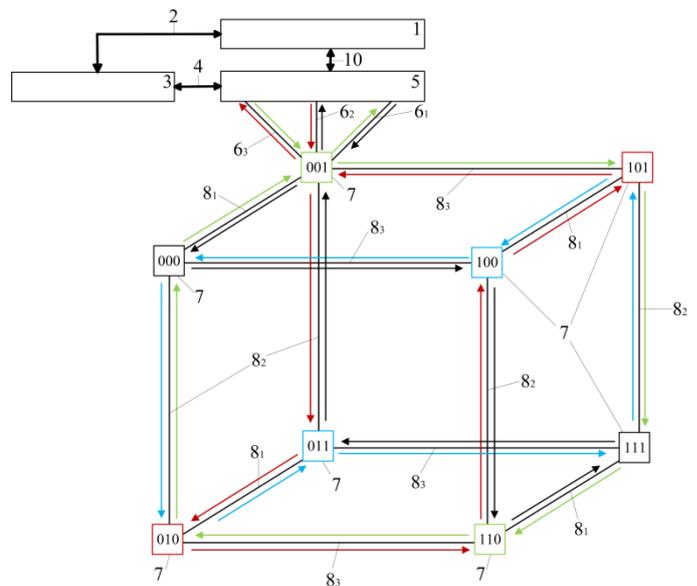


Fig. 2. Structure of a  $\Gamma_3$  photonic computer

Each node in the  $\Gamma_3$  topology represents a processor element associated with a vector  $\vec{\sigma} = (\sigma_3, \sigma_2, \sigma_1)$ , where  $\sigma_i \in \{0,1\}$ ,  $i=1,2,3$ .

The edges connecting the nodes of the  $\Gamma_3$  topology correspond to channels 8 denoted by  $8_1, 8_2, 8_3$ .

In addition, the processor elements are connected to device 5 by channels 6<sub>1</sub>, 6<sub>2</sub>, 6<sub>3</sub> that are external for  $\Gamma_3$ . In Fig. 2, channels 6<sub>1</sub>, 6<sub>2</sub>, 6<sub>3</sub> are shown only for the processor elements at the node (0,0,1).

For  $\Gamma_3$  and a route length of 3 [10] we have four conflict-free sets: ((000),(111)), ((001),(110)),

((010),(101)) и ((100),(011)). Their nodes and the conflict-free routes connecting these nodes in Fig.2 are represented by black, green, red and blue arrows, respectively. The arrows point in the data flow direction.

Tables 1 and 2 show the vectors  $\vec{\sigma}$  that correspond to the processor elements, from which information is delivered by conflict-free routes through channels 81, 82, 83 at time steps  $t=0,1,\dots,6$ , for the elements (0, 0, 0) and (0, 0, 1).

The MLS of the algorithm for the expression  $A=a+(b+c) \times d$  of the form  $Aabc+d \times + :=$  is presented in Fig. 3, and the processor graph is shown in Fig. 4. It consists of vertices corresponding to processor elements  $(\sigma_3, \sigma_2, \sigma_1)$  at time steps  $t$  and denoted by  $(\sigma_3, \sigma_2, \sigma_1)_t$ . The vertices are connected by edges from channels 8 belonging to the conflict-free routes.

The identifiers are distributed as follows: the

**Table 1.** Information ingestion in processor element (000)

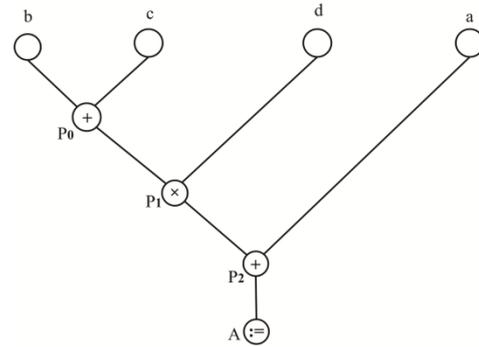
t	8 <sub>1</sub>	8 <sub>2</sub>	8 <sub>3</sub>
	001	010	100
0	000	000	000
1	001	010	100
2	011	110	101
3	111	111	111
4	110	101	011
5	100	001	010
6	000	000	000

**Table 2.** Information ingestion in processor element (001)

t	8 <sub>1</sub>	8 <sub>2</sub>	8 <sub>3</sub>
	000	011	101
0	001	001	001
1	000	011	101
2	010	111	100
3	110	110	110
4	111	100	010
5	101	000	011
6	001	001	001

identifier of the operation  $\{\times\}$  and the values  $b$  and  $d$  are loaded at the node  $(0,0,1)_0$ , the values  $A$ ,  $a$  and  $c$ , at the node  $(0,1,0)_0$ , and the identifiers of the operation  $\{+\},\{+\}$  and  $\{:=\}$ , at the node  $(1,0,0)_0$ . These characters are transmitted from device 5 through channels 6 as

shown in Fig. 4. The remaining elements receive the  $\emptyset$  characters.



**Fig.3.** MLS of the algorithm to compute the value of  $A=a+(b+c) \times d$  of the form  $Aabc+d \times + :=$

In particular, the route of the identifier  $A$  at the time steps  $t=0,1,2,3,4$  shown in Fig. 4 includes the nodes  $(0,1,0)_0, (1,1,0)_1, (0,1,0)_2, (0,1,1)_3, (0,0,1)_4$ . The routes of the  $\emptyset$  characters are represented partly as dashed lines.

Replacing the graph vertices with instructions, we obtain a program for the photonic computer. The program consists of instructions to be executed at each time step  $t$  by each processor element through each channel.

We distinguish three instruction execution phases:

- phase  $\varphi=1$ ; ingestion of information from channels by processor elements;
- phase  $\varphi=2$ ; execution of actions prescribed by the operation code by the processor elements;
- phase  $\varphi=3$ ; transmission of information from the processor elements to receivers in other processor elements.

Table 3 shows the program of a  $\Gamma_3$  photonic computer derived from the processor graph shown in Fig. 4. In each column  $(\sigma_3, \sigma_2, \sigma_1)$  for  $\varphi=1$ , the notation  $x/y$  means that the identifier  $x$  is received at the time step  $t=y$  through the channel  $8_i$  by the processor element  $(\sigma_3, \sigma_2, \sigma_1)$ ;  $\varphi=2$  indicates that the operation is completed by this processor element at the time step  $t=y$ , the dash means no operation; for  $\varphi=3$ , the notation  $x/y$  means that the identifier  $x$  is transmitted at the time step  $y$  through the channel  $8_i$  to the processor element  $(\sigma_3, \sigma_2, \sigma_1)$ ; the line in the notation  $x/y$  has the same color as the route, by which the information is received or transmitted.

Details of program conversion from high-level languages and machine representation are irrelevant to the present work. One can use any known tools to optimize the conversion and the resulting machine code.

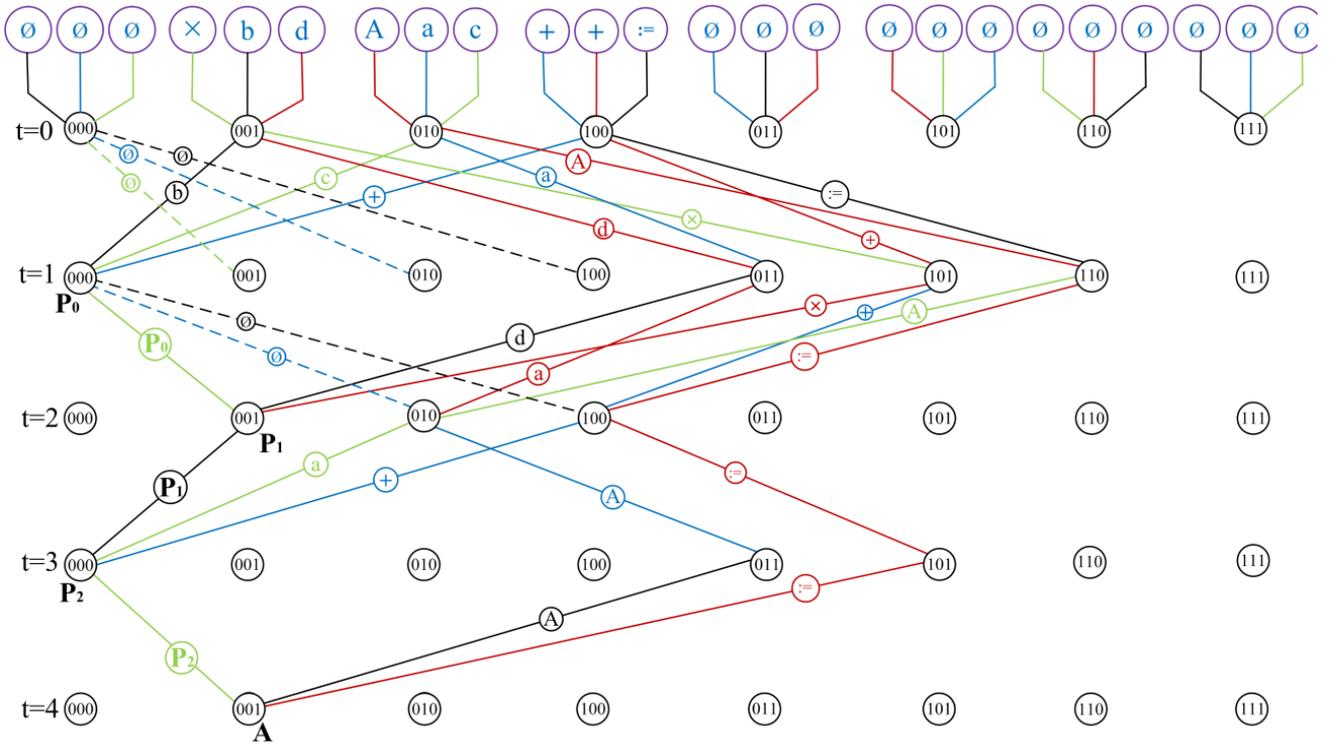


Fig.4. A processor graph to compute the values of  $A=a+(b+c)\times d$  of the form  $Aabc+d\times+:=$

Table 3. Program of the photonic processor for computing the value of  $A=a+(b+c)\times d$  on the form  $Aabc+d\times+:=$

t	$\phi$	000	001	010	100	011	101	110	111
		$\delta_1 \delta_2 \delta_3$							
0	1	$\emptyset \emptyset \emptyset$ 0 0 0	$\{\times\} b d$ 0 0 0	$A a c$ 0 0 0	$\{+\} \{+\} :=$ 0 0 0	$\emptyset \emptyset \emptyset$ 0 0 0	$\emptyset \emptyset \emptyset$ 0 0 0	$\emptyset \emptyset \emptyset$ 0 0 0	$\emptyset \emptyset \emptyset$ 0 0 0
	2	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -
1	1	$b c \pm$ 1 1 1	$\emptyset \emptyset \emptyset$ 1 1 1	$\emptyset \emptyset \emptyset$ 1 1 1	$\emptyset \emptyset \emptyset$ 1 1 1	$a d \emptyset$ 1 1 1	$\pm \emptyset \times$ 1 1 1	$\emptyset := A$ 1 1 1	$\emptyset \emptyset \emptyset$ 1 1 1
	2	$+ b c P_0$	- - -	- - -	- - -	- - -	- - -	- - -	- - -
2	1	$\emptyset \emptyset \emptyset$ 2 2 2	$P_0 d \times$ 2 2 2	$a \emptyset A$ 2 2 2	$\pm := \emptyset$ 2 2 2	$\emptyset \emptyset \emptyset$ 2 2 2	$\emptyset \emptyset \emptyset$ 2 2 2	$\emptyset \emptyset \emptyset$ 2 2 2	$\emptyset \emptyset \emptyset$ 2 2 2
	2	- - -	$\times P_0 d P_1$	- - -	- - -	- - -	- - -	- - -	- - -
3	1	$P_1 a \pm$ 3 3 3	$\emptyset \emptyset \emptyset$ 3 3 3	$A a \emptyset$ 3 3 3	$:= \emptyset \pm$ 3 3 3	$\emptyset \emptyset \emptyset$ 3 3 3	$\emptyset \emptyset \emptyset$ 3 3 3	$\emptyset \emptyset \emptyset$ 3 3 3	$\emptyset \emptyset \emptyset$ 3 3 3
	2	$+ a P_1 P_2$	- - -	- - -	- - -	- - -	- - -	- - -	- - -
4	1	$\emptyset \emptyset \emptyset$ 4 4 4	$P_2 A :=$ 4 4 4	$\emptyset \emptyset \emptyset$ 4 4 4	$\emptyset \emptyset \emptyset$ 4 4 4	$\emptyset \emptyset \emptyset$ 4 4 4	$\emptyset \emptyset \emptyset$ 4 4 4	$\emptyset \emptyset \emptyset$ 4 4 4	$\emptyset \emptyset \emptyset$ 4 4 4
	2	- - -	$:= A P_2 A$	- - -	- - -	- - -	- - -	- - -	- - -
	3	$\emptyset \emptyset \emptyset$ 5 5 5	$A - - -$	$\emptyset \emptyset \emptyset$ 5 5 5					

## 5 Implementation Features of a Photonic Processor

### 5.1 Communication topologies

A photonic processor is a multiprocessor system composed of processor elements connected by communication channels.

One can use any known [10] communication topology, like a 1D torus, a 2D torus, a 3D torus, a hypercube ( $H_m$ ,  $\Gamma_n$ ), a full crossbar (N), a star (S), or a fully-connected tree (F), implemented based on q-port switches. Selection criteria include the values of  $D_k$  (see Table 4), which is the number of routes between two processor elements located at a distance  $k$  (measured by the minimum number of edges between them), and the values of cardinality  $C$  and number  $G$  of conflict-free sets (see Table 5) estimated relative to the system diameter, which is the minimum value of  $k$  sufficient for connecting any two elements of the system composed of  $\omega$  processor elements.

**Table 4.** Number of routes of length  $k$

System		1D	2D	3D	$\Gamma^n$	N	F
$D_k$	2	$2 \cdot 2^{k/2}$	$4k^2 \cdot 2^{k/2}$	$k!$	1	$(q/2)^{k-1}$	

**Table 5.** Cardinalities and numbers of conflict-free sets

System	1D torus	2D torus	3D torus	$H^m$ $m = 2^{r-1}, r=0,1,2,\dots$	$\Gamma^n$	N S	F
C	2	$2\sqrt{\omega}$	$\frac{2}{\omega^3}$	$\frac{\omega}{2}$	$\frac{\omega}{2}$	$\frac{\omega}{2}$	$\frac{\omega}{2}$
G	$\frac{\omega^2}{2}$	$2\sqrt{\omega} \cdot 2\sqrt{\omega}$	$\frac{2}{\omega^3} \cdot 2\omega^3$	$(\frac{\log \omega}{\log \log \omega}) \frac{\log \omega}{2} \cdot 2^{\frac{\omega}{\log \omega}}$	$\binom{n}{m} \cdot G(H^m)$	$2^{\omega-1}$	$\frac{\omega}{2q^2} 2^q \cdot 2^{\frac{\omega}{2k}}$

The values of  $D_k$ ,  $C$  and  $G$  have been calculated in [10]. They determine the system's communication power and, consequently, the parameters of the processor graphs and programs.

The most "primitive" topology is the 1D torus. The best values of  $C$  and  $G$  are shown by the N and S topologies having the highest hardware complexity. Reasonably practicable values of hardware complexity are demonstrated by the F and  $\Gamma^n$  topologies. These topologies have close values of  $D_k$ ,  $C$  and  $G$ .

The  $\Gamma^n$  topology has a convenient conflict-free routing mechanism and is formally represented by Boolean functions; it offers topological redundancy facilities to replace failed processor elements without any impact to the system topology and computation [10].

### 5.2 Interaction with the environment

Communication between the photonic processor and its environment should take as little time as possible. This becomes possible if every processor element of the

photonic processor has access to the environment; for example, if there is a dedicated electronic processor to perform storage and communication functions.

Electronic processors corresponding to the photonic processor elements can be connected by communication channels into a multiprocessor system, in particular, having the same topology as the photonic processor. This makes it possible to use the same representation for both electronic and photonic parts of a photonic computer.

### 5.3 Hardware of the photonic processor

Hardware of the photonic processor is a full-function set of OLGs to perform any arithmetic and logic operations. Such a set includes (but is not limited to) A, OR and NOR logic gates [9].

The gates can be realized using various effects, like interference [5, 6], nonlinear effects, etc. [3]. The gates are passive and do not require additional pumping energy. All operations are performed only with the optical form of information, and the operation time is determined by the time of light transmission through the OLG.

## 6 Performance Estimates

Let us estimate the peak logic performance  $\pi \rho$  of a photonic computer, which is the number of logic operations performed by the computer per unit time.

Incoming light of power  $P_\Sigma$  is distributed among all processor elements and their input OLGs.

The light power ingested in the waveguides of the input OLGs is  $P_1$ . In one processor element we have  $b = P_\Sigma / (\omega \cdot P_1)$  circuits of connected OLGs.

The total OLG waveguide length per circuit does not exceed  $L = (\mu/\Delta) \cdot 10 \lg(P_1/P_{\min})$ , where  $\mu \leq 1$  is the light ingestion efficiency of the waveguide,  $\Delta$  represents losses in OLGs and waveguides,  $10 \lg(P_1/P_{\min})$  is the power budget [11], and  $P_{\min}$  is the minimum value of OLG output power.

For up-to-date technologies,  $\mu = 0.9$  and  $\Delta = 0.1$  dB/cm [14].

The values of  $P_{\min}$  are calculated based on the amount of energy  $Q_{\min} = N \cdot E$  delivered to the OLG

input, and the pulse length  $\tau$  to deliver this energy. Here,  $N$  is the number of photons per pulse,  $E = h \cdot \nu / \lambda$  is the photon energy,  $h = 6.62 \cdot 10^{-34}$  J·s is the Planck constant [12],  $\nu = 2 \cdot 10^8$  m/s is the speed of light in the waveguide,  $\lambda$  is the wavelength.

We assume that the input pulse length and the operation time of an interference-based OLG are the same and equal  $\tau = D/\nu$ , where  $D = j \cdot \lambda$  is the OLG length, and  $j$  is an integer.

We obtain  $P_{\min} = N \cdot h \cdot \nu^2 / j \cdot \lambda^2$ . Suppose  $\rho_c$ ,  $\rho_u$  and  $\rho_\Sigma$  are the numbers of OLGs in one circuit, one processor element and the photonic computer, respectively. We have  $\rho_c = L/D$ ,  $\rho_u = b \cdot \rho_c$  and  $\rho_\Sigma = \omega \cdot b \cdot \rho_c$ .

It is evident that  $\pi\rho = \rho_\Sigma / \tau$ . We get  $\pi\rho = (P_\Sigma / P_1)(\mu / \Delta) \cdot 10 (\lg P_1 / P_{\min}) \cdot \nu / j 2 \lambda^2$ .

It follows from the latter that if  $\lambda$  decreases  $z$  times,  $\pi\rho$  increases  $z^2$  times (other multipliers having the same values).

For example, for  $\lambda = 1530$  nm,  $j = 50$  and  $N = 104$ , we obtain  $P_{\min} \approx 2 \cdot 10^{-3}$  W. Assuming  $P_1 = 20 \cdot 10^{-3}$  W,  $\mu = 0.9$ ,  $\Delta = 0.1$  dB/cm, we find  $\rho_c = 1.2 \cdot 10^4$  pcs.,  $\rho_u = 0.75 \cdot 10^7$  pcs. For  $P_\Sigma = 100$  W and  $\omega = 8$  we obtain  $\rho_\Sigma = 6 \cdot 10^7$  pcs. and a peak logic performance of  $\pi\rho \approx 1.8 \cdot 10^{20}$  operations per second.

The value of  $N$  depends on the OLG implementation technology and medium parameters (absorption coefficient, temperature, etc.) These aspects are beyond the scope of this paper. We only note that at small values of  $N$ , for example,  $N = 2$ , one can encounter effects specific to quantum computing [15].

OLGs within a processor element can be grouped into various devices performing arithmetic operations (which are sets of logic operations) according to prescribed algorithms.

As we know [9], multiplication of 64-digit floating point numbers requires  $\sim 27K$  logic operations. Consequently, the peak arithmetic performance of the photonic computer under the above conditions will be  $\pi\rho = 0.7 \cdot 10^{17}$  multiplication operations per second, which is 104 – 105 faster compared to up-to-date electronic computers with the same power consumption [16].

We assume that the width and thickness of the substrate does not exceed 100 microns. Eight processor elements of the photonic computer under consideration will occupy at most  $0.5 \cdot 10^{-3}$  m<sup>3</sup>.

This size of a structural component allows removing 100 W of heat power by conventional air cooling.

For vast values of the parameters, the prospects of achieving which are not discussed here, namely,  $P_\Sigma = 10^8$  W,  $\lambda = 1.0$  nm,  $j = 10$ ,  $N = 20$  and  $\nu = 3 \cdot 10^8$  m/s, we obtain  $\pi\rho = 1030$  operations per second. Who, where and when will be able to achieve this value?

## 7 Conclusion

The efficiency of information processing in the optical form is provided by joint application in the photonic computer of:

- passive optical logic gates;

- computing discipline based on the availability of operands (data flow);
- conflict-free information processing algorithms for processor elements networked into a multiprocessor system.

Photonic and electronic computers compute the same classes of problems.

The above analysis and the resulting estimates demonstrate that a photonic computer working on light with a wavelength of 1530 nm can perform 104 – 105 times faster than up-to-date electronic computers with the same power consumption.

Light waves of different wavelengths do not interact with each other. With a proper OLG implementation, several computations represented by light waves of different wavelengths can be run on a single photonic computer.

## References

1. L.V. Tarasov, *Laser Physics, 2nd ed.*, (Libricom, Moscow, 2010) [in Russian]
2. S. Binkley, Quantum Computing. ASCAC DOE US. April 5, (2016)
3. P.A. Belov, V.G. Bespalov, V.N. Vasiliev, et al., Optical processors: Advances and new ideas.
4. Russel John, High-enabled Microprocessor Holds Promise for Faster Computers. February 23, (2016)
5. A.O. Poluektov, An optical logic gate. RF Patent No. 20751061. Published 10.03.1997.
6. M. Papaioannou, E. Plum, Y. Valente, E.T.F. Rogers, and N.I. Zheludev, *APL Photonics* **1**(9), 090801 (2016)
7. S.A. Stepanenko, *Doklady Mathematics*, **96**(2), 522-527 (2017)
8. S.A. Stepanenko, A photonic computer. RF Patent No. 263969. Priority February 20, 2017 [in Russian].
9. B.Ya. Tsilker, S.A. Orlov, *Organization of electronic computers and systems* (St.Petersburg, PITER, 2004) [in Russian].
10. S.A. Stepanenko, *Multiprocessor systems of supercomputers. Efficiency scaling* (Moscow: FIZMATLIT, 2016) [in Russian].
11. A.A. Verbovetskii, *Fundamentals of optoelectronic communication system engineering* (Moscow: Radio i Svyaz', 2000) [in Russian].
12. B.M. Yavorskii, A.A. Detlaf, A.K. Lebedev, *Handbook of Physics for Engineers and Undergraduate Students* (Moscow, Oniks, Mir i Obrazovanie, 2008) [in Russian].
13. V.V. Voevodin, *Mathematical Models and Methods in Parallel Processes* (Moscow, Nauka, 1986) [in Russian].
14. N.V. Nikanorov, S.M. Shandarov, *Waveguide Photonics* (ITMO, St.Petersburg, 2008) [in Russian].

15. M. Nielsen, I. Chang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000, 2001)
16. Timothy Prickett Morgan, Nvidia Rounds Out Pascal Tesla Accelerator Lineup. June 20, (2016)