
3D ICS with Optical Interconnections

Edward G. Kostsov, Sergey V. Piskunov and Mike B. Ostapkevich

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/47872>

1. Introduction

Today's microelectronics, whose main driving force of development has always been the needs in computing devices, has achieved exceptionally great results over the last decade.

The exponential growth in performance of microelectronic devices, predicted in 1965 by Gordon Moore and formulated as an empirical law, was sufficiently accurate for more than 45 years: the computing power of single-chip microprocessor-based systems increased almost by a factor of four every three years. Simultaneously with the improvement of parameters of ICs the performance of supercomputers has increased. The performance has increased both due to a reduced duration of a cycle and due to pipelining and parallelization of computations. Currently, petaflops computers are built on the basis of 10^5 - 10^6 single-chip processors with a clock frequency of 10 GHz.

At the same time, the analysis of the element base of microelectronics has shown that the main obstacle to improve the performance of computer devices is the problem of connections, both on the surface of individual substrates, and between them. The area occupied by conductors is about 70% of the overall area of a crystal itself in modern VLSI. The total length of conductors exceeds the linear dimensions of a crystal by more than a thousand times. The energy spent on charging the conductors is 60-70% of all energy losses. Placement of the conductors on the surface of a crystal requires significant technical efforts, while the limit value of capacitance of a conductor, which is 10^{-11} F/m., is attainable [1]. On schematic level, there is also a problem of RC propagation delay in the connections. And consequently, the velocity of signal propagation in ICs is much lower than that of light (5-30 times) depending on the degree of integration.

Thus, the main source of increasing the performance of computers at the previous period of their evolution, which is concluded in increasing the performance of a single gate and the degree of integration, is nearly exhausted. Thus, it can be stated that the clock frequency of computers is determined by circuit limitations, rather than by physical and technological ones.

No less complicated problems arise while exchanging information between processor chips. With an increase in their size the number of inter-chip connections increases as well. And this reduces the clock speed by almost an order of magnitude. With decreasing the cycle duration, the above-mentioned connections play an increasingly significant role.

From the stated-above it can be concluded that the solution to the problem of connections and thus further improvement of the performance of computer devices can be obtained by the following ways:

- increasing the degree of parallelization of information processing at all levels down to elementary operations (a maximum depth of parallelization) and increasing the decentralization of functions of storage, control, and data processing, and transition from long logical links within the surface of a chip to the local relations between the neighboring gates, i.e. by transition to a VLSI with a homogeneous structure;
- transition from connecting chips placed in a plane to 3D packs of VLSIs from these chips and to their vertical interconnections.

It follows from general physical considerations that a simple way of introducing the "third dimension" into the structure of communications is using optical links. Such channels have the following advantages:

- A high degree of parallelism of the information transfer from plane to plane makes possible to use highly parallel algorithms for processing (down to elementary operations), and thus to create high-performance computing.
- Capability of optical synchronization allowing delivering a synchronization signal to any point of a chip practically without delay, e.g. from one light source outside an IC.
- Optical communications channels are free from parasitic effects of the mutual influence because of the neutrality of photons. With increasing the clock frequencies (especially in the gigahertz frequency range), the advantages of optical signals grow as capacitive coupling between electronic conductors with increasing frequency grows as well.

There is always an interest in the use of optics for high-performance computer devices. Recently, a special attention of developers of such devices has been drawn to the development of a new element base, creation of optoelectronic integrated circuits (OEIC), for the organization of inter-IC and inter-processor connections in order to attain high throughput and low power consumption [1-4].

The optical communication bandwidth over short distances within a few millimeters already competes with electric conductors with sufficiently low energy consumption, up to 1 pJ/bit/m or less. The placement of optical communication channels directly on the chip surface can significantly reduce this bandwidth. The reports are already known about the first experimental data on providing the density of the bandwidth 37 Gbit/s/mm² and higher by using a switched CMOS pair of a vertical-cavity surface-emitting laser (VCSEL) and a CMOS-compatible avalanche photodetector (CMOS-APD) system. The frequency of switching light sources VCSEL, electro-optic modulators MQW based on superlattices GaAs / GaAlAs, InGaAsP, etc. as well as that of their corresponding photodetectors has already

reached values of $20 \div 50$ GHz, which in the short term, can grow up to $70 \div 80$ GHz. The level of technology of constructing field-programmable smart-pixel arrays and FP-SPA systems based on the use of optical channels in free space, is now able to provide the exchange of information between the two surfaces at a rate exceeding 10 Tbit/cm².

Currently, however, optical communication channels are used only for the information transfer problems. Their application directly to the microelectronic structures and in the channels of information processing virtually does not evolve. The main reason is that the size of components of gates (modulators and light sources) due to the wave nature of the optical signal is much larger than that of the active elements of modern microelectronics. Entirely optical computers performing massively parallel computations typically contain rather large elements: lenses, shadowgrams, spatial light modulators, etc. and cannot be created using microelectronics technology.

At the same time, it may also be noted that with the development of nanophotonics, the advent of the light sources and receivers with nanometer dimensions (see, e.g. [5-7]), the optical channels could be used not only for the exchange of information between ICs, but also for the information processing, with the 3D logic devices. Thus, the construction of optoelectronic schemes, whose implementation fits well into the existing technology of building semiconductor circuits, is actual.

The *objectives* of this paper are: a) to consider the possibility of creating high performance and manufacturable multi-layer (3D) chips in which a processing of information and its mass exchange between the layers is performed using optical communication channels and whose parameters are compatible with those of microelectronic circuits, b) to represent the algorithmic and computer tools providing the design and research of simulation models of such 3D algorithms and structures with massive parallelism that are oriented to an optoelectronic implementation.

The principles of the gate, where logic signals are represented by the presence or absence of a light flux driven by an electric field are described in the paper. The possibility of the physical implementation of 3D logic circuits based on such elements is analyzed and its advantages are discussed. The methods and means of the algorithmic design of 3D ICs with a homogeneous structure for execution of algorithms with fine-grain parallelism are described. These tools include a formal model of fine-grained computing, Parallel Substitution Algorithm (PSA) and a simulation system (WinALT), which is based on PSA and provides the construction of models of devices with a 3D architecture. Features of the system are demonstrated by constructing models of 3D optoelectronic matrices for parallel data processing. Matrices can be characterized by a high performance, simplicity of cells, homogeneity and simplicity of the topology. Simulation models help in acquiring an objective view of the complexity and performance of the matrices and confirm the reasonability of the transition to 3D VLSI in order to overcome the problem of connections that arises in modern 2D VLSI. The WinALT system is available in Internet. The expediency of transforming WinALT to an online environment for supporting the simulation of 3D computational structures with the users' participation in the network (virtual) community is discussed.

2. The physical basis of implementation of 3D optical logic circuits

2.1. Optoelectronic gate

Let us consider features of constructing a universal gate, which is based on the modulation of a light flux by an electric field [8-11], and when only a single transition of energy (between light and electric signal), the source of a light flux is shared by many gates. The scheme of such a gate is presented in Figure 1. It has the following main components: a light modulator LM controlled by the electric field, EC converter of a light signal to an electric signal (e.g. a photoelectric transducer), energy storage or a load element LE (capacitance for dynamic circuits). The element controls the intensity of the luminous flux transmitted by LM, and with the help of the light flux entering the optical input of EC.

The most optimal physical and technical solution, as follows from [8-11], is obtained when the light modulator is electro-optical, the energy converter is photovoltaic and the energy storage is electrostatic. Such a decision is mainly due to the consideration of energy. In this sense, the use of magneto-optical and acousto-optic light modulators is limited by a high energy consumption and by the complexity of the direct energy conversion (photomagnetic or photoacoustic).

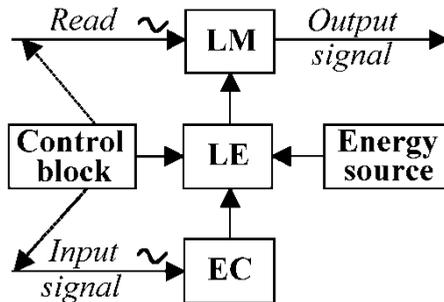


Figure 1. Flowchart of universal gate

The state of a gate is determined by the ability of the light modulator to pass a light flux coming on its input (state "0" or "1"). This state is unambiguously related to the amount of energy stored, which, in turn, is determined by the intensity of the light flux entering the energy converter (entrance gate). According to Figure 1 a gate in the dynamic mode operates as follows:

- supply of energy from the energy source (voltage pulse) to LE at the time t_0 (operation "erasing information");
- supply of a light flux on the input of the transmitter PE at the time $t_0 + \Delta t$ (operation "information recording"), where Δt is the duration of a cycle;
- supply of a light flux on the input of LM at the time $t_0 + 2\Delta t$ (operation "read information").

Assume that the modulator lets the light flux pass when the amount of energy stored in LE, and does not let the light pass otherwise. Let us also assume that when EC receives a light

flux composed of the streams X_1, X_2, \dots, X_n , such energy in the load element is released. Then the light flux on the output of the modulator Y (at the moment of reading) will be the Pierce logical function:

$$Y = \overline{X_1 \vee X_2 \vee \dots \vee X_n}.$$

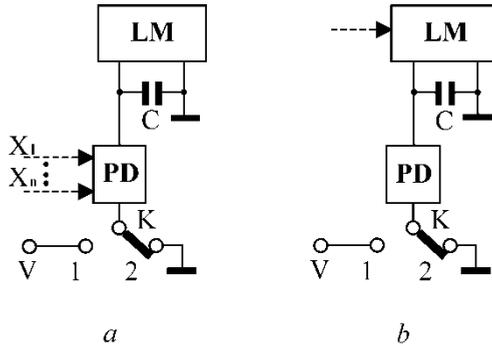


Figure 2. Universal gate with optical connections

This function forms a complete basis of Boolean functions, so the element proposed can be considered universal.

The electric functional scheme of the gate is shown in Figure 2, where K is the key, V is the power supply voltage. The gate implements the Pierce function and it operates by cycles according to the following general description.

cycle 1 - "erasing information", key K is in the position 1, and capacitance C is being charged by the source when the photodetector FP is illuminated.

cycle 2 - "storing" K is in position 2 (see Figure 2, a), the light flux with intensity J_1 or J_2 (depending on the level of illumination corresponding to signal "1" or "0", i.e. whether there is at least one of X_i not equal to "0") comes to the optical input of the gate (photodetector FP). According to the level of illumination the capacitance C is discharged with the time constants $R_T C$, or $R_C C$, and the $R_T C \gg R_C C$, where R_T is dark resistance of FP , and R_C is its resistance under illumination.

cycle 3 - "reading", key K is in position 2 (see Figure 2, b) and the light signal passes to the optical input of the modulator. The light flux corresponding to "0" light is obtained from the output if signal "1" arrived at the input in the second cycle. And vice versa, signal "1" with a high intensity of light is on the output of LM if there was signal "0" in the previous cycle.

The storage time of information received by the element is proportional to the value of $R_T C$. However, we can create a dynamic memory by inclusion of two gates so that the output of one of them be connected to the input of another and vice versa, as shown in Figure 3. Its storage time is limited only by the time of maintaining the voltage on the power supply.

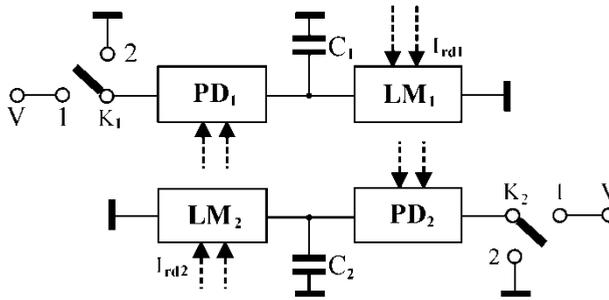


Figure 3. A dynamic memory cell

When constructing digital devices based on these optical gates the interlayer connections are implemented by placing pairs of "modulator - photodetector" in different layers-planes so that the optical output light modulator of one gate is geometrically aligned with the photodetector optical input of another gate.

2.2. A sample of constructing a dynamic memory cell

Such a cell is intended for storing information in the electric form on capacitors and transferring information between cells by pulsed optical signals, while a direct transfer of charge does not take place. The scheme of the cell is depicted in Figure 3.

Assume that a bit of information (e.g. "0") is written in the upper gate 1. The switch K_1 was shot to pin 2 and PD_1 was closed in order to do so.

Cycle 1. "Erasing" of information in lower gate 2 by charging the capacitance C_2 under lit PD_2 and the switch K_2 shot to pin 1 of the power source with voltage V .

Cycle 2. The reading of information from element 1 by the light signal I_{rd1} . The switches K_1 and K_2 are in position 2. The output signal from LM_1 is the input writing signal for element 2.

Cycle 3. The "erasing" of information in element 1. The switch K_1 is in position 1 and the capacitance C_1 is connected to the power source with voltage V and is charging.

Cycle 4. The reading of information from element 2 by the light signal I_{rd2} . The switches K_1 and K_2 are in position 2. The output signal from LM_2 is the input writing signal for element 1.

The information in a cell is represented by a pair of light signals ("0", "1") or by voltages taken from pins of capacitances. It can be kept as long as needed as a result of iterative repetition of cycles 1-4. If "1" was written to a cell in the initial state, it would have kept a couple of "1", "0".

Similarly, other schemes of optical elements can be built to implement the basic logic functions: repeater, inverter, AND, NOT-AND, OR, NOT-OR, sum by modulo 2, implication. The analysis of transfer characteristics for such elements is presented in [10].

The described principle of implementation of inter-layer communications allows creating functionally flexible devices by replacing electric logical communications by the optical ones and by using 3D structures of gates. A computational device would contain a minimum number of elements and electric connections if the following rules of its creation are followed: parallel electric circuits are used to supply power to elements and parallel logical links are done using optical channels.

An illustration of the functionality of a logic gate with optical connections is presented in [10, 11] using samples constructing 1D and 2D shift registers, switch and matrix processor for parallel image processing. The processor consists of a control block and a program controlled cellular automaton. The control block stores a program and fetches its instructions. The cellular automaton performs information storage and processing [11]. A cell in cellular automaton transforms information by changing its own state and those states of its neighbors. Any transformation is represented as a sequence of elementary transformations. Each elementary transformation is defined by the contents of a command that comes to a cell from the control unit. The transformation of information in a cellular automaton is performed simultaneously by all cells.

The following conclusions can be drawn from the analysis of the functioning of the described devices (primarily the matrix processor), the specific features of their design and comparison with microelectronic devices, capable of performing similar functions.

1. A cell of matrix processor contains significantly less elementary components (estimated by two orders of magnitude) as compared to conventional chip with similar functional capabilities.
2. The total number of cycles spent on execution of logical operations required for image processing is also smaller, at least by a factor of N (where N is the number of lines in the image). The time of the complete image processing does not depend on its size. In particular, a very small number of clock cycles is required to execute such operations as selection of a contour image, noise filtering, noise filtering with masking, extension of lines, etc., which are quite complex for electronic circuits. And the number of cycles remains unchanged when image dimensions increase.
3. The area occupied by the connections is 7 - 8% of the total area of a substrate, as shown by the analysis made. The described devices have a higher volume density of placement of elements, the reliability of connections, noise immunity, and ultimate manufacturability. In addition, the design of an optical VLSI is considerably simpler, since there is no need to take a complex configuration of interconnections into consideration.

A breadboard construction of a cell of optic dynamic memory is created in order to demonstrate the possibility of practical implementation of optoelectronic gates [11]. It consists of two optoelectronic logic gates. Lithium niobate crystals are used as optoelectronic modulators.

The study of the dynamic memory model has made possible to draw the following conclusions:

- transfer of information from one element to another can be done without loss of information an unlimited number of times;
- the use of threshold photodetectors allows reducing of the operating voltage and of energy required for switching a gate down to acceptable levels for microelectronic implementation.

3. The Cellular technology constructing 3D logical structures

Efforts to create optoelectronic circuits have given impetus to the development of algorithmic and software tools that aid in solution of the problem of design and study of 3D (multilayer) digital structures, focused on the use of optical interlayer connections.

These tools map a primordial parallel algorithm of a problem solution onto an architecture with a massive spatial-temporal parallelism. The orientation to constructing structures of models that consist of huge sets of rather simple and homogenous computing devices (cells), mainly with local links, and placed in a 3D space is their basic property.

The considered tools include:

- a formal model of fine-grain (cellular) computations, which is called the Algorithm of Parallel Substitutions (PSA) and which serves as the basis of a method for synthesis of parallel architectures;
- a simulation system of parallel computational processes, which is used for the construction, debugging of models of 3D logical structures as well as for the extraction of the characteristics from these models.

A detailed description of the cellular technology is given in [11].

3.1. PSA: A generalized model of fine-grain computations

Conceptually PSA unites in itself a substitutional character of Markov's algorithm [12] and spatial parallelism of a cellular automaton [13] basing on an associative mechanism of application of operations, which is common to both of them. PSA represents a "true parallelism" of computations, when all the allowable operations are executed at each step for all the available data. The main idea of PSA is concluded in the following three statements:

- The processed information is presented in the form of a cellular array, which is a set of cells. Each cell is data (a bit, a character, a number, etc.) with its name (its "location" within array, which is an element of a set of names M) in the array. A set of data belongs to a certain finite alphabet A .
- The algorithm is defined by a set of parallel substitutions. They have left-hand and right-hand sides (left and right parts). The expression in the left part generates cellular arrays, one for each cell name in the processed cellular array. If processed arrays contain one or more such arrays, then the substitution is applicable. Its execution means that a certain "base" part of the found array generated by its right part is replaced by array generated by its left part for the same cell name. Its execution means that a certain

"base" part of the found array is replaced by the array generated by the right part of the parallel substitution for the same cell name.

- The process of computations is iterative. At each iteration, all the substitutions applicable to the processed cellular array are executed. The computation is over when no substitutions applicable to the array obtained at the previous iteration were found. This array is the result of work of PSA.

Executing the substitution in the form of replacement of one cellular array by another permits representation of such replacement as replacement of one spatial image by another. This is rather essential for a visual construction of computer-aided model of optoelectronic logical structure and visual representation of computational process of such a model, which is distributed in time and space.

A formal description of PSA is presented in [14]. Let us demonstrate the idea of PSA by a simple example. Let us consider PSS Φ_σ for adding many non-negative binary numbers as a sample.

Let $A = \{0, 1\}$, $M = N^2$, $N = \{0, 1, 2, \dots\}$. A transformed cellular array is a 2D rectangular table whose cells are numbered according to left coordinate system (i is abscissa, j is ordinata). The name of cell m is a pair of values i, j . The digits of numbers are kept in the strings of the table. The least significant bits occupy the right-most column. PSA Φ_σ consists of two parallel substitutions:

$$\Phi_\sigma = \left\{ \begin{array}{l} \Theta_1^\sigma : \{(1, \langle i, j \rangle)(1, \langle i, j+1 \rangle)(0, \langle i+1, j \rangle)\} * \{(0, \langle i, j-1 \rangle)(0, \langle i+1, j-1 \rangle)\} \rightarrow \\ \{(0, \langle i, j \rangle)(0, \langle i, j+1 \rangle)(1, \langle i+1, j \rangle)\}; \\ \Theta_2^\sigma : \{(1, \langle i, j \rangle)(0, \langle i, j+1 \rangle)\} * \{(0, \langle i, j-1 \rangle)\} \rightarrow \{(0, \langle i, j \rangle)(1, \langle i, j+1 \rangle)\}. \end{array} \right.$$

The left part of the substitution is to the left of the arrow, while its right part is to the right of the arrow. Shift functions are written in angle brackets in the definitions of substitutions Φ_σ . These functions set the location of cells of the left and right parts of a substitution as related to each other. A cell description is confined in parentheses. Cell states are shown to the left of commas. The "base" parts of the left parts of substitutions are at the left of asterisks. When particular values of the pair i, j are substituted into the shift functions, the cellular arrays associated with this name are obtained.

The description of substitutions allows their geometric representation depicted in Figure 4, a. In this case, the left and right parts of commands are defined by templates, and the search of occurrences of the left parts of substitutions is done by shifting their templates above cells of the table along axes. One step of transformation of the source table containing numbers 9, 15, 5 is presented in Figure 4, b. After carrying out four steps the result is placed in the top string, while the rest of them contain zeros.

The expressive capabilities of PSA augment if an alphabet A is extended by introduction of *variable* and *functional* symbols. An alphabet A serves as a domain for variables symbols and as a range for functional symbols. In the case of graphic representation of substitutions,

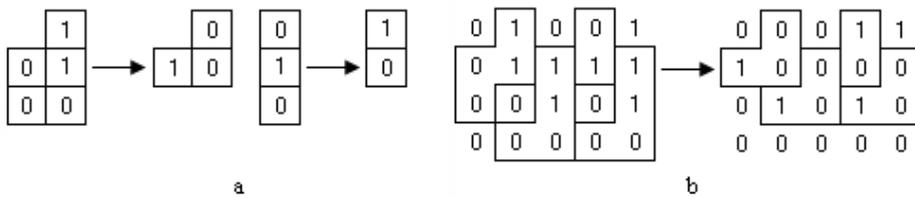


Figure 4. Cellular adder: a – graphic representation of PSA commands, b – one step of execution of algorithm

variable symbols can be written into cells of templates of the left part of substitution, while the functional symbols can be in the cells of the right part. Such a substitution is called *functional* substitution. When an alteration of a state is performed for a cell located in the processed array under a template cell of the right part of a certain command containing a functional symbol, it is not specific data from the template cell, but a result of evaluation of a certain function that is written into this cell. The states of cells in the processed array that are below the cells of a template of the left part can be arguments of such functions. Using the functional substitutions helps: a) in reducing PSA notation to a considerably more concise form in theoretical issues, b) in practical issues to represent rather complicated devices such as ALU by a single cell.

3.2. A simulation system of 3D digital structures

3.2.1. Overview of the system

A physical and technical rationale for the construction of multilayer optoelectronic structures is given In Section 2. Evidently the construction of real 3D structures requires a large amount of work associated in the first place with selection of an optimal kind of structure from the standpoint of technical parameters among a big number of possible variants. A manual solution of such a task is virtually impossible. Thus, a computer-aided tool must be available for the research and development of models of 3D structures. Such an instrument was built and it is called WinALT [15]. The user's interface of the system coincides with the standard user's interface in Windows applications. A simulation model is represented by a project that contains a number of sub-windows. Each sub-window can hold graphic or text objects of a model. Creation and editing of graphic objects are carried out by means of toolbars, menus and dialog windows. The system is freely distributed. System's site [16] contains a section "installation", which includes manuals on installation, uninstallation and system's distributive package. An open architecture of the system enables the user to participate in extension of the system functionality.

3.2.2. Adequacy of the system language to the problem domain

The WinALT was developed simultaneously with computer-aided models of parallel algorithms and structures under their strong influence. This influence has manifested itself in the WinALT system by a wide employment of visualization tools both for supporting the

construction of parallel algorithm descriptions (graphic representation of objects corresponding to cellular arrays, the left and right parts of substitutions) and for their simulation (capability to view the dynamics of application of each substitution). What is more, a special attention was given to the development of tools that provide the visualization of 3D (multilayer) objects and that allow viewing the transformation of cell states in any layer of a 3D cellular array. This is due to the orientation of the development of computer-aided models to fine-grain architectures that are promising for the implementation in the form of multilayer VLSI. The simulation language contains three parts.

The first part of the language is designed to describe parallel computations in the form of parallel substitutions. It is fully based on PSA.

The objective of the second part of the language is the description of sequential computations. This part is essentially based on Pascal. It provides statements for the description of simulation program structure, control operators, assignment operator and subroutine call by name or by reference. These statements can be used in a model program for the description of sequential control when needed. These means can be used in a model program for the description of sequential control, when it is needed, for the definition of functions that describe cell states and, also, for the construction of such service functions within a model program as menu definition, graph drawing or initial data input.

The third part of the language provides importing libraries into a model program. These are **dll** libraries written in C/C++ and embedded into the simulation system. This helps in enriching the functionality of simulation tools to suit the user.

Let us describe the first and the third parts in greater detail.

The Description of the Parallel Part of Simulation Language. This part has a clear division into graphic and analytical subparts. Graphic objects are cellular arrays and templates. An image of an object is composed of color cells located along horizontal and vertical axes and, also, along the axis that goes from the user to the screen. The origin in the template is called its *center*.

A color is used to visualize a cell's state. Its state can belong to any cell's data type supported by the library of data formats (see below). A name can be assigned to a cell in addition to a typed value as an additional property. It has to be unique within the scope of one cellular object. This makes possible to implement functional substitutions in WinALT. There is a special neutral **void** state of a cell (depicted by a diagonal cross on a color background).

A parallel substitution in WinALT simulation language is set by a bunch of operators **in-at-do**. The names of cellular arrays and templates can be used as parameters in these operators. In the case of one-block structure of a device (a single cellular array), a parallel substitution is described as follows. The parameter of **in** operator is the name of a processed cellular array. A name of template of the left part of substitution is a parameter in **at** operator, while that of its right part is a parameter in **do** operator. The execution of substitution is done in two phases. During the first one, the center of a template of the left part is moved along the axes in a processed cellular array, and all its occurrences in this array are marked accurate to the

empty cells. At the second phase, the states of cells of the processed cellular array in all these occurrences are replaced by the states of cells from the template of the right part also accurate to the empty cells. Taking multi-block structure of the device into consideration means the following. The parameter in operator **in** is a list of names of cellular arrays placed in brackets and separated by commas. The lists in operators **at** and **do** are arranged similarly, but instead of names of cellular arrays the names of templates are used in them. Combining patterns in the list means that the movement of patterns in the images of their corresponding cellular arrays is coordinated by substitution of the same set of coordinates into the centers of all the templates of the operators **at** and **do**. The coordinates of cells of the cellular array that is in the head of the list **in** operator form these sets. Such substitutions are called vector substitutions and allow describing parallel transformations of information in compositions of cellular arrays.

A bunch of operators **in-at-do** and a description of a function serve as an analog of functional substitution. Operator **at** contains the names of templates, in which some cells are named. A function uses these names as input and output variables. The name of a function is used as a parameter in operator **do**. A functional substitution can also be a vector one.

A synchroblock **exhaust-end** (or shortly **ex-end**) is the main structure for definition of an algorithm of a device operation. This block implements an iterative procedure of PSA application for the composite operators describing parallel substitutions that it contains. In addition, there are two more kinds of synchroblocks that were introduced: **clock – end** (**cl – end**) and **change – end** (**ch – end**). The first one executes its substitutions a number of times specified as its parameter. The second one executes its body only once.

Remark. The combination of parallel and sequential parts of the language is attained by the possibility to use operators from the sequential part in synchroblocks. Let us also note that nested synchroblocks can be in WinALT simulation language. The described capabilities allow constructing any parallel-sequential compositions of synchronous transformations of cellular arrays.

Model program. The structure of a model program is quite conventional. It consists of a list of libraries imported to a program (using operators **use**, **import** and **include**), declarations of constants, variables and cellular objects, procedures, functions and the main operator block. The main operator block is placed in operator **begin-end** brackets and contains operators of the first and second parts of the language. A program may include comments, which can be placed in braces. A project can contain any number of simulation programs. They are capable of interacting with each other if necessary.

The third part of the language is based upon a set of WinALT libraries. The functionality of the system is extended by means of external modules. These modules are represented by Windows **dll** files. These external modules contain the interface functions that are used in versatile simulation models. The external modules form several groups that are called libraries. Some of them are briefly described below.

The library of data formats eliminates limitations of a data type that can be represented by cells in a cellular array. The library contains modules for representations of cellular arrays with

integer cells (**int8**, **int16**, **int32**, **uint8**, **uint16**, **uint32**), bit cells (**bit**), float cells (**float**) and others. Some external formats are supported by the modules of library, such as **bmp** raster graphics format. The assignment of **default** type for a cellular object means that any of its cells can have any of the above-mentioned formats. The latter can be used for the representation of heterogeneous cellular objects. In GUI, the type of a cellular object can be selected in a combo box within the dialog window of the new object creation.

The library of language functions provides the ability to use such functions in simulation programs as functions of object management (creation, deletion, modification or size alteration), GUI functions (construction of dialog windows and data input based upon them), mathematical functions (**sin**, **cos**, **atan**, **cosh**, **log**, **j0**), console I/O functions (**WriteLn**, **ReadLn**), file I/O functions (**fopen**, **fgets**, **fread**, **feof**, ...) and miscellaneous functions such as **max**, **min**, **null**, **typeof**, **StringLength**, **Time**. Operator **use** activates the modules of this library in a simulation program. A typical module of this library is a **dll** written in **C** or **C++**.

The library of visual modes provides a customizable visualization of a cellular array and its cells. A cell state can be visualized e.g. by color, a directed arrow or by a number or by their certain combination. A 3D cellular array can be shown as a deck of layers or as layers unrolled in a line or in a grid.

New external modules can be added to any of these libraries. Such modules can be created, for example, using Microsoft Visual Studio. Their source texts can be either borrowed from a provided sample, written from the scratch, or taken from an existing library of functions (e.g. ANSI C runtime library).

3.3. Constructing simulation models of a family of optoelectronic matrices in WinALT

3.3.1. Mapping digital schemes onto a customizable cellular automaton

The cellular automaton with Margolus (further CA) neighborhood [13], in which setting cells for execution of a certain set of elementary transformations of information, serves as a logical basis for the construction of a family of optoelectronic matrices. The CA is a double layer automaton. Its first layer functions in the same way as the Margolus automaton, being a rectangular matrix of cells. Let us split the matrix to blocks of 2x2 cells. Let us call it E-partition. Let us split the matrix again to blocks that are shifted as related to the blocks of E-partition by one cell along the vertical and horizontal axes. Let us call it O-partition. Each cell of the first layer can be in one of the three states: "white", "gray" or "black". Let us call this layer *informational*. The layer under the informational one is called *control* layer. Its E-partition coincides with that of the informational layer. This layer keeps the table of settings the control layer of CA. The size of a cell in the table coincides with that of a block. The CA is shown in figures as a sweep of two layers in a plane. The informational layer is on the left side and the control layer is on the right side. The blocks of E-partition are limited by solid lines. The blocks of O-partition are limited by dashed lines. An elementary transformation of information performed in the informational layer is a parallel substitution. Its left and right

parts are matrices with 2×2 cells. These matrices are composed of white, gray and black cells. The elementary transformations performed in a CA are enumerated. Setting a CA consists in writing numbers of those elementary transformations that can be performed in the block of informational layer above a cell into that cell. Digital (combinatory) schemes of different kinds (adder, multiplier, etc.) can be implemented by setting a CA. A source combinatory scheme is initially transformed in a way that any of its gates has no more than two inputs and two outputs. All the cells of the informational layer are white in its initial state. The picture of a scheme to be imitated in CA, which is made by gray cells in informational layer of CA is called *image* of digital scheme. The signals transformed in a digital scheme are depicted by black cells in its image. The states of cells in each block of the information layer of a CA form a certain picture called *image* of a block.

The graphic images of commands of parallel substitutions are depicted in Figure 5, a. A number of a command is shown above an arrow. The transformations of information in Margolus cellular automata are performed by alternating E- and O-partitions [13]. The alternation of partitions is substituted by alternation of two groups of shifts of the image of simulated scheme as related to the fixed control layer in a CA. A group of shifts set into correspondence with E-partition is called *E-group*. A group of shifts set into correspondence with O-partition is called *O-group*. The introduction of a setting for CA allows reducing the number of parallel substitution commands that are executed by a single block to two. The command, whose number is specified first in a cell of the settings table of the control layer, can be executed when E-group is active, while the command, whose number is listed second, can be executed when O-group is active. A command is executed if its left part coincides with the image of a block in the informational layer. The execution of a command sets its right part as the image of a block.

Possible variants of signal transmission in the image of a digital scheme (horizontal and diagonal transmissions, branching and crossing of signals) and the indications to the functional elements (AND gate is denoted by the sign '&', OR gate is denoted by '1', addition by modulo 2 is denoted by '=1', half-adder" is depicted at end) are shown in Figure 5, b.

When E-group of shifts is active, the commands are executed for the source disposition of the image of a digital scheme, then for the image shifted by one row of blocks from E-partition up as related to the source image, and then one row of blocks down as related to the source image. The shifts are introduced in order to provide the execution of commands imitating the operation of gates and crossing of signals transmission channels in the image of a digital scheme. When O-group of shifts is turned on, commands are executed for the image obtained by shifting the source one by one row down and one column to the left.

The horizontal transmission of a signal is simulated by commands 1, 2. The diagonal transmission is performed by commands 3, 4. The signal branching is done by 5, 6 and signal crossing - by 8, 10, 15. The operation of AND gate is simulated by the two sets of commands: 9, 16, 17 and 12, 16, 17, because the result on the output of the gate can be recorded either in the right top cell or in the right bottom cell. of the block. Similarly, the

simulation of OR gate is done by commands 7, 8, 9 and 10, 11, 12 and the simulation of addition - by modulo two is done by commands 7, 8, 18 and 10, 11, 18. A block of E-partition that does not contain white cells can simulate a gate operation not only with one output, but also with two outputs. Thus, the operation of half-adder is simulated by commands 7, 8, 12. The capability to simulate digital schemes with two inputs and two outputs allows performing both the signal transfer, e.g. by one of the diagonals, and a logical transformation of this signal and another signal. For example, a signal transition from the right bottom cell into the right top cell of a block along with performing AND operation with signals from the left column of cells in the block with writing the result to the right bottom cell of the block is simulated by commands 8, 15, 16.

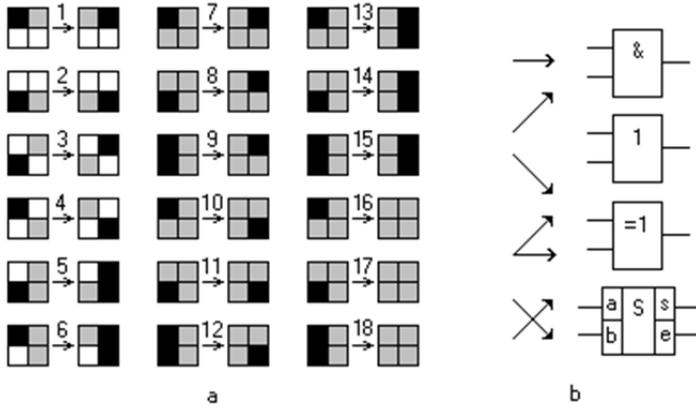


Figure 5. Graphic representation of parallel substitution commands, a - images of commands, b - symbols of signals and functional elements

Let us demonstrate an image of a full adder in CA as a simple sample of what was stated above. It is presented in Figure 6.

Let us comment on this figure. The numbers of commands listed in Figure 5, a are used in the settings table. An image of a digital scheme with mnemonic specification of the functions of its composing blocks except those imitating the transfer of signals under O-partition is given in Figure 6, c in addition to the image in Figure 6, a, b. A sign of operation or an arrow in the image of scheme denotes a cell where result of transformation or data transfer is placed. Such a representation is rather easy to grasp and is introduced solely for the purpose of the reader's convenience. This helps in comparing a combinatory scheme and its image without the table of settings.

Usually, the signal transfers will be omitted when using such a kind of representation of a digital scheme unless that hampers the perception of the image of a scheme. Let us describe the resulting image of a digital scheme. Each of blocks (2, 2) and (3, 3) corresponds to a half-adder, which is a composition of AND gates and of addition by modulo two. Blocks (4, 2) correspond to OR gate.

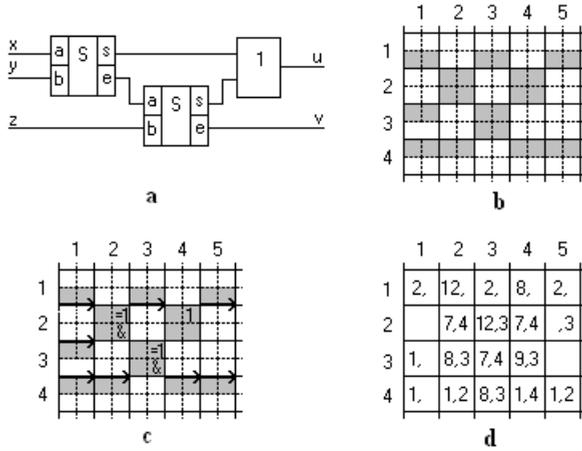


Figure 6. Full adder: a - a combinatory scheme, b - an image of scheme, c - a correspondence of blocks, d - the settings table

Some or all of the gray cells in the first column of CA have to be altered to black ones in order to introduce input signals to a full adder.

A more complicated and realistic sample is presented in Figure 7. The following heuristic criteria are selected for the estimation of quality of a particular mapping. A mapping is optimal if there is at least one such chain from the inputs of a scheme to its outputs, that is composed only of blocks that imitate gates connected with other by corner cells. The image of an eight-bit pyramidal adder [17] transformed into a scheme, in which each gate has two inputs and one or two outputs is presented in Figure 7.

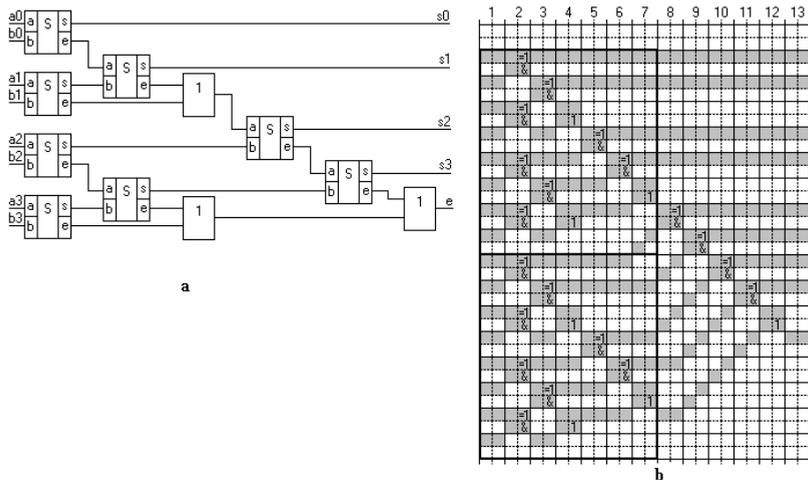


Figure 7. A pyramidal adder: a - the combinatory scheme of a four-bit adder, b - a concise image of eight-bit adder scheme

The image of a four-bit adder, which scheme is depicted in Figure 7, a, is a source fragment for its construction. The images of four-bit adders are contoured by a bold line in Figure 7, b. The rest of the scheme image provides daisy chaining of the selected images of adders. The image of an adder presented in Figure 8, b meets the heuristic criteria. The proposed way of constructing an image of a digital scheme with a specified bit width by connecting its homogeneous fragments of smaller bit width, for each of which the best image is already found, can be used for other kinds of schemes as well.

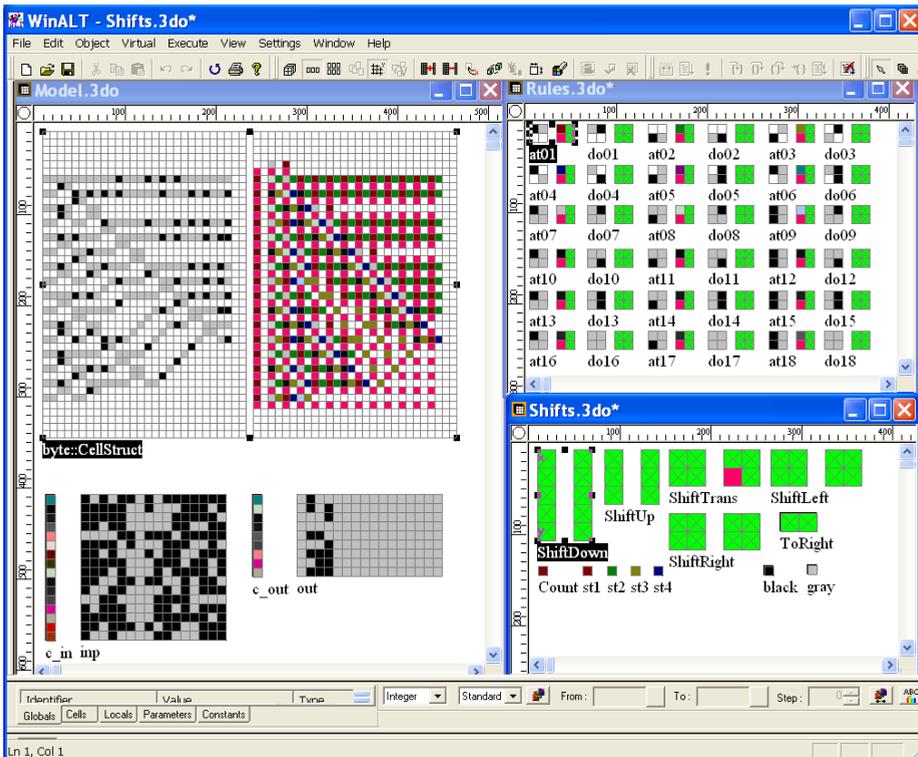


Figure 8. A project window of simulation model of CA that implements an eight-bit adder

From a logic standpoint, a fine-grain structure (a micro-pipeline) is implemented by setting a CA up. This structure simulates a simultaneous operation of many copies of the same digital scheme. Each copy (let us call it a virtual digital scheme) performs a transformation of its own data set. A *stage* of a micro-pipeline forms a vertical column of blocks in E-partition of matrix. Each stage is a "cut" of image of a virtual scheme in a certain phase of transformation of the relevant information. This feature of matrix makes possible to obtain new results in each cycle. A cycle of micro-pipeline operation includes transformation and transfer of data from one stage to another. It means that a cycle includes an execution of the both groups of shifts. A model is constructed in WinALT in order to evaluate the correctness of an eight-bit adder implemented as CA.

3.3.2. CA simulation model implementing eight-bit adder

The screenshot of a window of this project is presented in Figure 8. This window contains sheets with all the graphic objects of the model.

The sweep of a double layer cellular array **byte::CellStruct** is presented at sheet **Model.3do**. The upper layer containing the image of an eight-bit adder is presented at the left. The bottom layer is shown at the right. It contains a computer-aided representation of the settings table of an adder. The division of array layers into blocks is done by setting the cells of the bottom layer with both even coordinates into a state dedicated for this purpose. The numbers of substitution commands are represented by color in the settings table. The numbers of commands are distributed by the block cells as follows: the right and left top cells of block keep the numbers of the first and second commands respectively. The rest of cellular arrays shown in this window are auxiliary. Array **inp** keeps a set of input data. The columns of array **inp** are written in each cycle to the leftmost column of the informational layer of array **byte::CellStruct** using the array **c_in** and become the values of outputs of an adder. Similarly, the results of computations from the very right column of the informational layer of array **byte::CellStruct** are transferred to array **out** that stores the results of pipeline computations. Double layer templates are presented in sheet **Rules.3do**. The templates named **ati** and **doi** are put in correspondence with the left and right sides of substitution command with the number **i**. The upper layer of template **ati** coincides with the left part of the substitution number **i**, while that of template **doi** coincides with the right part of that substitution. The left upper cell of the bottom layer of template **ati** contains number **i** of substitution command. The number is represented by a color. A color of the left bottom cell of the same layer coincides with that of those cells in the bottom layer of array **byte::CellStruct**, which provides its partition into blocks. All the other cells of the both templates are empty (diagonally crossed). The templates used in substitution commands of the procedure imitating clocking of CA are presented in sheet **Shifts.3do**. A cycle of a model execution is divided into the four stages. At each of them parallel substitution commands kept in the settings table are executed:

- above the source image at the first stage;
- above the image shifted two cells up from the source one at the second stage;
- above the image shifted two cells down from the source one at the third stage;
- above the image shifted one cell down and one cell right from the source at the fourth stage (the command interleaving is done in the settings table during this stage: odd and even columns are swapped).

The operation of gates of a scheme is simulated at the first three stages of cycle. The data transfer between gates is simulated at the fourth stage. The cellular image and the settings table return to the initial ("unshifted") state at the end of a cycle.

Let us briefly present main procedures of simulation program. The description of parallel substitution commands is kept in procedure **mainProc**, while the shifts performed at the each stage of a cycle are defined in procedure **shiftImage**. The body of procedure **mainProc** contains operator **in** with parameter **byte::CellStruct** and eighteen bunches of **at-do**

operators. The parameters of operators **at** and **do** in the i -th bunch are the templates named **ati** and **doi** respectively. The shifts of image of the digital scheme with respect to settings layer and the interleaving of substitution commands is done in procedure **ShiftImage** by vector functional substitutions. Cellular arrays **byte::CellStruct** and **Count** listed in round brackets in the first operator **in** form a composition, which has the following meaning. The changes in array **byte::CellStruct** happen only when the unicellular array **Count** is in a certain state. For example, if the state of **Count** coincides with that of **st2**, the shift of the image of a digital scheme is done by one row of blocks down from its source placement. This shift is performed by the function **fShift** that uses the values of variables **x** and **y** from template **ShiftDown** in its operator **y:=x**. The operators **at-do** in the second operator **in** alter the state of **Count** in order to set phases of cycle.

3.3.3. A transformation of CA into optoelectronic matrix

Let us list those operations converting optical signals, which are to be carried out in an optoelectronic matrix to implement the above-mentioned CA. The operations have to provide the following for each block of CA: a) a comparison of the image of the left part of a command, whose number is written in the settings table, with the image of block in the image of a scheme, and b) if they coincide, replacement of the image of such block by the image of the right part of a command. Let us demonstrate that an execution of such operations is possible in a four layer matrix (called **S**). The images of left parts of substitution commands are kept in its first layer. The second layer contains the image of a digital scheme. The images of comparison schemes are in the third layer. The images of right parts of substitution commands are in the fourth layer. Matrix **S** is built using the basic gate rather schematically and without details of implementation. All the essential physical elements: modulators, photodetectors, memory cells are considered to have a size equal to one cell. The reasons for that are as follows: 1) matrix **S** built in such way can be easily turned into a data object when it is simulated, 2) the obtained matrix **S** has rather generalized form that makes possible to specify its versatile representations taking into consideration real sizes of its elements later, because these sizes are not known a priori and depend on physical principles of construction of the elements, technology, and the application domain.

The cells of informational layer of CA have three states. But the basic gate is binary. Thus, a transition must be done to binary encoding of the states of matrix **S**. For example, an encoding of white, gray and black cells presented in Figure 9 can be chosen.

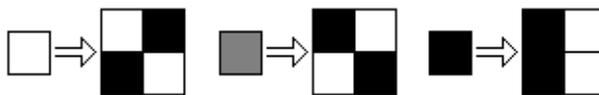


Figure 9. Binary encoding of CA states

A white cell denotes a memory cell containing one, an opened modulator or an opened photodetector. A black cell denotes a memory cell containing zero, a closed modulator or a closed photodetector.

Let us commence the construction of matrix *S* using its second layer that contains an image of a digital scheme in a binary encoding. An image of each block in CA informational layer is composed of modulator states. Memory cells control the states of modulators (opened, closed). Assume that each cell has two outputs. Modulators are connected to both of them. Let us consider that each memory cell has two inputs, each of which is connected to a photodetector. Let us also assume that the signals from the inputs of a cell appear at its output with a fixed delay. A line of cells is composed of such elements. The first and second cells are modulators. The third cell is a memory cell. The fourth and fifth cells are photodetectors. A memory cell operates in two phases. If it contains one at the first phase (the phase of comparison), it opens the first modulator and closes the second one. And otherwise, it closes the first modulator and opens the second one. At the second phase (phase of recording) if the fourth photodetector is opened, and the fifth one is closed, one is written to memory cell. Otherwise, memory cell resets to zero.

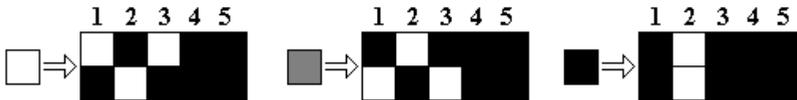


Figure 10. Binary encoding of cell states in the second layer of optoelectronic matrix

An image of the second layer of matrix *S* when it is ready for the phase of comparison is obtained by replacing each cell in the information layer of CA by a group of cells according to the substitution rules in Figure 10. Thus, a block of optoelectronic matrix of size 10×4 cells composed of four lines is put in correspondence to each block of the informational layer of CA.

Remark. Gray cells are also used in the transition to binary encoding, but they denote "void", absence of any hardware.

Let us perform a partition into blocks that have the same sizes and placement as in the rest of matrix layers.

Let us construct the first layer of matrix *S*. Unlike lines of cells form the second layer, the cell number four contains a memory cell and the fifth cell is gray. A memory cell is connected to modulators just as the memory cell occupying the third cell. The basis for setup of the first layer is the settings table of the control layer of CA. The setting of layer starts with encoding of its E-partition, i.e. the cells in columns 3 and 8 of the layer are set in such a way that the image formed by the states of modulators would coincide with the left part of that command, whose number is written first in the cell with exactly the same placement in the settings table of CA. The encoding for O-partition is embedded into the encoding of the layer that was just obtained. For this purpose binary codes set into correspondence to white, gray and black cells of the left part of the second command are written into the fourth column just as it is done for the cells of the third column in Figure 10.

The third layer of the matrix is constructed as follows. Black cells of a single block in the columns 1, 2, 6, 7 denote closed photodetectors. Black cells in the columns 4, 5, 9, 10 denote

closed modulators. The rest of columns are empty. They are composed of gray cells. Pairs of photodetectors in rows 1, 2 and 3, 4 of the columns 1, 2 and 6, 7 are connected by OR scheme (parallel). Then all the OR schemes formed by pairs of photodetectors are connected sequentially forming AND scheme. A parallel assembly of all modulators of a block is connected as its load. The scheme obtained in the third layer allows to detect the situation when the images of blocks coded by the states of modulators located in the first and second layer one below another coincide, and in the case of full coincidence to prepare a substitution of the image of block of the second layer by the right part of command kept in the fourth layer.

The construction of the fourth layer can be done in two stages. First, a layer that is exactly the same as the first one for the right parts of commands is built. Then the columns in each block are transposed in the following order: 5, 4, 3, 1, 2, 10, 9, 8, 6, 7.

A fragment of matrix S presented in Figure 11 illustrates the procedure described above. The fragment is built for such a block of CA, whose settings table has a cell that contains the numbers of substitution commands 7, 4. The fragment is in such a state when E-shift is done and it appeared that the image of the block in the informational layer of CA coincides with the image of left part of substitution command number 7. A polarized light comes perpendicularly to the first and last layers of the matrix, respectively, from above and from below. The phase of comparison of images of blocks in the layers 1 and 2 of matrix S and detection of their coincidence is shown in Figure 11, a. The modulators in columns 4, 5 and 9, 10 are opened. The phase of writing new states of memory cells of the second layer using photodetectors connected to their inputs is depicted in Figure 11, b.

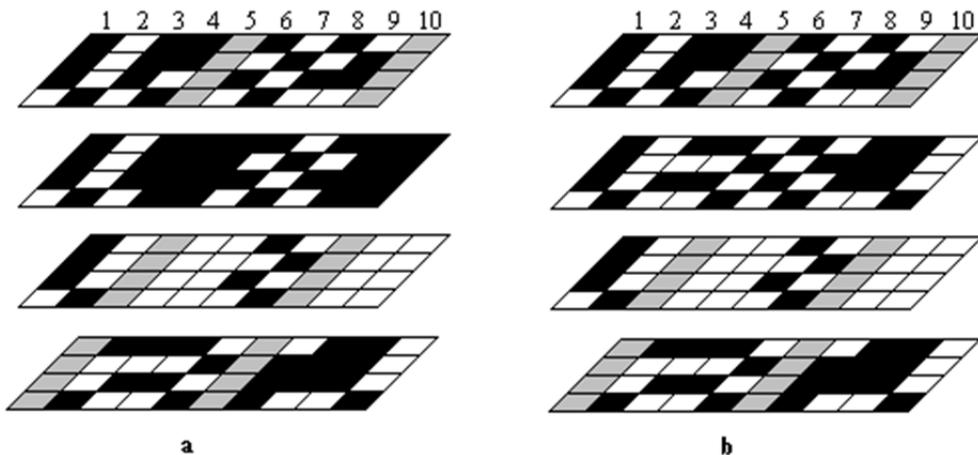


Figure 11. A fragment of the matrix S : a - detection of matching images, b - the phase of setting of new cell states in the second layer.

After a certain fixed delay, memory cells of the second layer set new states for modulators on their outputs. A fixed delay can be implemented, for example, by division of a memory cell into two elements: main and buffer. Modulators are connected to the outputs of the

main buffer. Photodetectors are connected to the inputs of a buffer element. One cycle of state transition from a buffer element to a main one serves as a delay. Then the next shift of the image of a digital scheme is performed and the phase of comparison of image blocks can be done again.

It can be stated that CA turns into a simple homogeneous optoelectronic matrix S . It is a device that consists of four layers, which contain microscopic passive sources of light (optoelectronic modulators) and light detectors, forming a regular flat structure. The control of light sources and detectors is performed using electronic memory cells. Memory cells form a 2D shift register in order to provide setup of memory cells for storing the image of a scheme and alternation of commands. The obtained matrix is reconfigurable. It can be setup to imitate any combinatory scheme. The logical operations in the matrix (a comparison of the two codes and writing a code to memory) are preformed optically and their corresponding electronic gates simply do not exist. The proposed matrix has high performance. Indeed, after the pipeline has entered in the steady state, each next result is produced on the outputs of the matrix in each cycle.

3.3.4. *The family of optoelectronic matrices*

Matrix S can serve as a basis for the construction of a family of similar matrices. Let us outline the possible ways of their construction. The circuitry of matrices may have rather different implementations. For example, one can construct a matrix of static memory elements (as in the previous section), but it is also possible to construct a matrix with dynamic memory elements. Another way of binary encoding of white, gray and black cells can be chosen. However, we consider more profound transformations of the original matrix. And here the following options are available.

The first way. A number of layers can be reduced to two in a matrix. Let us first note that the need to alternate substitution commands in the control layer is induced by the necessity to execute commands in O-shifts. A modification of CA is proposed in [18] that eliminates the group of O-shifts and as a result the need to alternate commands in the settings layer. Let us select such a CA as a basis for the construction of a new optoelectronic matrix $S1$. The selection of such CA means that the memory cells in columns 4 and 2 can be erased in the lines of blocks of the first and fourth layers along with their modulators and photodetectors.

Let us introduce an additional limitation. Let us consider that a CA is setup once and only for implementation of a single digital scheme. Selecting a constant setting leads to the fact that there is no need to change the state of modulators in the first and fourth layers of the matrix S in the process of its operation. In its turn, this means that the memory cell can be removed, an open modulator can be replaced by a transparent plate, and a closed one by a dark plate. But then the first and fourth layers of the matrix S can be removed and the modulators in the second and third layers under dark plates can be masked or removed from the substrate. The execution of such a procedure can be demonstrated in greater details using masks in substitution command 1 (Figure 5) taken as a sample. A set of masks is presented in Figure 12.

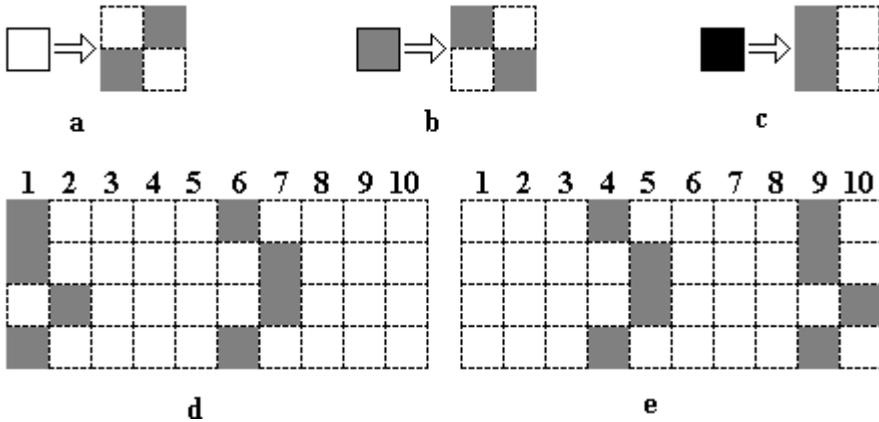


Figure 12. Masks: a - of a white cell, b - of a gray cell, c - of a black cell, d - the left part of substitution command 1, e - the right part of the same command

Masks **d** and **e** are built using masks **a**, **b**, **c**. Mask **d** is superimposed on those blocks of the second layer, which are assigned to execute command 1. The modulators under gray cells are removed. Other elements of a block under white cells remain intact. Mask **e** is applied to such blocks of the third layer, that are located under similar blocks of the second layer, and the same kind of actions is performed. As a result a computer simulation model that implements a full adder for such a matrix was built.

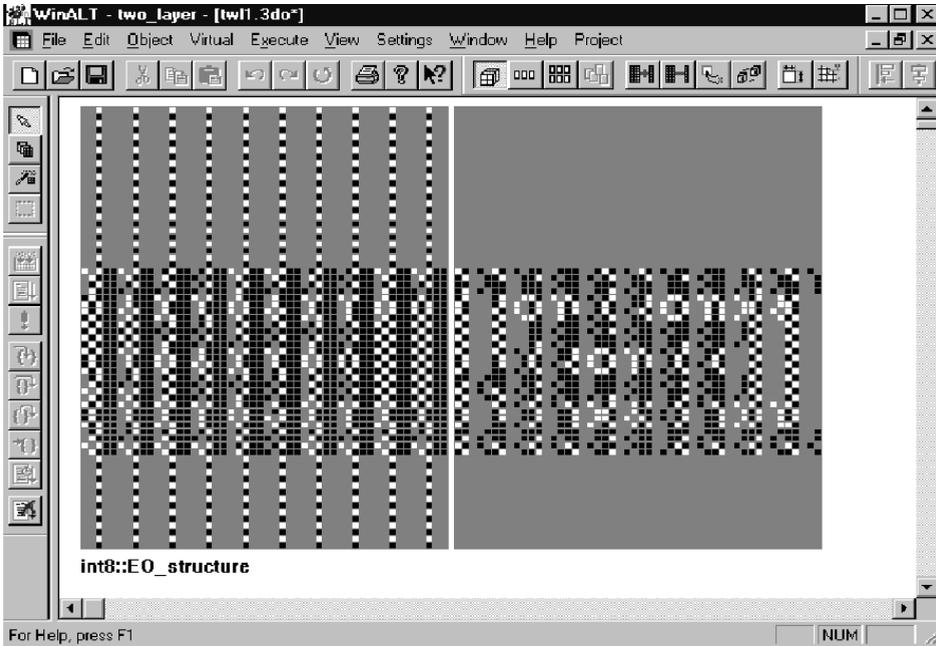


Figure 13. Computer image of a two layer optoelectronic matrix

A certain step of computations performed in a double layer matrix is presented in Figure 13. The obtained model is rather realistic. All the optical and electronic components are shown in Figure 13. By analyzing the screenshot, one can easily draw a conclusion that the first layer of matrix is composed of logically isolated shift registers. Each of their digits has optical inputs and outputs. The second layer composed of isolated single comparison schemes. This way can be considered as one with the minimum fraction of electronic components in the matrix. The double layer matrix and its simulation are presented in details in [18].

However, it is clear that when considering a practical implementation of optoelectronic devices, one has to take into account the achievements of modern microelectronics, where the actual geometric dimensions of the transistors (approximately 30 nm) have become significantly smaller than the sources of light, which can not be less than 0.3 microns due to the physical limitations. They become a weak element in terms of achieving high-density packaging of elements in a chip. Therefore, when constructing such a family of matrices, it is useful to consider the maximum use of electronic components. "Optical" components keep only their main function, which is a three-dimensional organization of the logical connections and processing of cellular (logical) information. The electronic components are then used to perform simpler operations that support a computational process, such as the shift and storage of information, including settings, etc.

The second way. One can augment the electronic part of the hardware of matrix **S** and turn it into the matrix **S2** with extended capabilities. Suppose that there are **k** ($k > 1$) four layer matrices and each **i**-th matrix is set to perform operation **O_i**. Assume that the sizes of the matrices and the location of inputs and outputs of simulated digital circuits are coordinated. Some of the considerations on how that can be obtained can be found in [19]. Let us combine all the electronic components of all these matrices in order to superpose them in a four layer matrix. It can be done the following way. A set of **k** memory cells is placed in layers 1, 2, 3 of the matrix instead of a single memory cell. Let us combine **k_i** cells into a 2D shift register set for execution of operation **O_i**. A decoder (named **D1**) is placed near each set of memory cells in the first and fourth layers. Similarly, two decoders are placed in the second layer (one **D1** and one that is different from **D1** is named **D2**). Decoder **D1** has one group of outputs and two groups of inputs. The states of outputs can be 0 and 1. They are connected to modulators of layer the just the same way as the outputs of memory cells in the source matrix **S**. The first group of inputs of decoder **D1** is for control. The codes of operations **O_i** ($i = 1, \dots, k$) come to it. The second group of inputs has **k** pairs of inputs. The pair of outputs of the **i**-th memory cell of a set is connected to the **i**-th pair of inputs. If an operation code **O_i** comes to the control input of a decoder, the states of the **i**-th pair of the second input group are sent to the outputs of decoder **D1**. Decoder **D2** is also controlled by operation codes. But its input group has two inputs, and it output group has **k** pairs of outputs. The inputs are connected to the outputs of photodetectors of the second layer the same way as the inputs of memory cells in the source matrix **S**. The **i**-th pair of outputs of decode **D2** is connected to the inputs of the **i**-th memory cell of a set. If an operation code **O_i** came to the control input of decoder **D2**, the states of inputs are sent to the outputs of **i**-th pair of decoder **D2**. Each set

of input data has to be accompanied by an operation code for such a matrix so as to specify what has to be done with these data. The transfer of operation codes from one micro-pipeline tier to another and their delivery to the decoders of the destination tier must be implemented. These actions can be performed for example by using additional shift registers that have to be inserted in the first, second and third layers of the matrix **S2**. As a result a matrix is built, in which one can change the image of a digital scheme and its setup table dynamically. That means that k different digital schemes can reside simultaneously in such a matrix. Thus from a logical standpoint the matrix is dynamically reconfigurable in the process of computations done by its pipeline. Matrix **S2** has undoubted advantages in comparison with matrices **S**, **S1**. The delays of its pipeline are virtually eliminated, because its control device would always "find" a data set and an operation to perform and deliver those to the inputs of matrix **S2**.

The list of matrices is not limited by the samples presented above. Let us briefly outline some more possible ways of constructing new modifications of matrices. For example, a dynamically reconfigurable matrix can be built on the basis of a two layer matrix. In order to make it reconfigurable, one has to introduce electronic masking control first of all. Then an additional hardware is to be added just as it was shown in the previous section. A group of E-shifts of a digital image can be eliminated in all the proposed matrices by increasing the vertical sizes of blocks that are compared in layers at least by a factor of three. This technique helps in shortening the duration of a micro-pipeline cycle. An assessment of creating vertical assemblies built, for example, on the basis of two-layer matrices, seems to be rather interesting. Double-layer matrices are interleaved with layers of light sources in such kind of assemblies.

And finally, let us mark one more important advantage of optoelectronic configurable matrices. Their application replaces the design (logical and physical) and production of digital circuitry by a programming of an already existing matrix. Indeed, in order to get a digital circuit in a matrix, one need only to write specially selected long vectors to its layers, using the shift registers, which are formed by memory elements of the matrix layers.

4. Conclusion

Two most pressing problems appeared in constructing computing systems: 1) an increase in the number and length of the connections inside 2D ICs, 2) a massive parallelization of computations.

Using a 3D (multilayer) IC structure is proposed in this chapter in order to solve these problems. Each of the layers, forming such an IC, has a homogeneous cell structure. A data exchange between these cells is performed by the means of local electronic (intralayer) and parallel optical (interlayer) communication channels. Data exchange between layers is combined with their processing. Cellular 3D ICs are oriented to computations with fine grain parallelism.

The physical and computing aspects of creating such ICs were considered.

The physical aspects include constructing an optoelectronic gate that allows combining single-layer ICs in the two-layer structures and an experimental test of its operability. The assessment of potential of project of a customized device, a matrix processor for the image processing from the standpoint of its performance and of its manufacturability permits to draw the following conclusions about the prospects of the chosen direction of constructing optoelectronic circuits.

Despite the fact that the size of one of the components (light modulator) of the optoelectronic logic gate is much larger than the active components of the modern microelectronics because of the wave nature of the optical signal, the performance and manufacturability of specialized devices of this type can greatly exceed similar characteristics of a purely electronic device. This is due to the fact that the number of elementary components in an optoelectronic device is significantly less than that in a purely electronic device with the same functionality. Reducing the number of components results from a local structure of logical connections and from the unidirectional light propagation (in electronic conductors a required direction of motion of the electrons is obtained by using a set of gates). There are virtually no intersections in optical conductors. This greatly facilitates the design of ICs and lets them be cheaper, more reliable and easily manufacturable. High throughput between layers by the means of optical channels gives a hope that an essentially greater performance could be reachable in comparison with purely electronic schemes.

Computer aspects of the work are devoted to the foundations of the algorithmic design of 3D ICs based on the formal model of fine-grain parallel computing, Parallel Substitution Algorithm, and to WinALT simulation system, presented in [15, 16]. A family of optoelectronic matrices was developed using WinALT. Comparison of two graphic images in all matrices, which is a main logical operation, is carried out optically exactly the same way as in devices built on the basis of optoelectronic gate. The functions of information storage are carried out electrically. High homogeneity, simplicity of topology and a low complexity of cells are the features of the matrix. A wide range of matrices was built. They vary by a number of layers, by functionality and by ratio of optical and electronic components. The proposed matrices can serve as an ALU basis in a general purpose CPU. The selection of a particular choice depends primarily on the kind of physical parameters, on which a designer can be oriented to, and on the kind of operations that a matrix has to perform. And this requires interaction between experts in different domains: physics, algorithm, programming, etc. The above suggests the following conclusions about the directions in which algorithmic design tools are to be developed.

The system's site must evolve into a fully functional online resource providing the ability to create a network (virtual) team consisting of specialists in different domains, and let them actively participate in a joint development of a 3D IC using the Internet.

The mapping of only one version of a fine-grain structure, a reconfigurable CA, to an optoelectronic structure is presented in the paper. There are many fine-grain structures and algorithms for different purposes, which may be of interest for optoelectronic implementation.

It is advisable to perform a constant replenishment of the collection of simulation models on the system's site both by its developers and by users. Also a collection of simulation models of optoelectronic implementations of fine grain structures and algorithms must be created. Let us also note the need to constantly replenish the system by new modules that extend its functionality with the emergence of new kinds of fine-grain structures and algorithms.

Constructing realistic optoelectronic structures requires constructing simulation models with huge amount of data and computations. It is expedient to launch such models on supercomputers. Thus, a WinALT subsystem must be developed for parallel execution of these models on clusters and supercomputers.

Author details

Edward G. Kostsov

Institute of Automation and Electrometry, Russian Academy of Sciences, Russia

Sergey V. Piskunov and Mike B. Ostapkevich

*Institute of Computational Mathematics and Mathematical Geophysics,
Russian Academy of Sciences, Russia*

5. References

- [1] Jin-Sung Y, Myung-Jae L, Kang-Yeob P, Woo-Young C. 10-Gb/s 850-nm CMOS OEIC Receiver with a Silicon Avalanche Photodetector. *IEEE journal of quantum electronics* 2012; 48(2) 229-236.
- [2] Young I A, Edris M, Liao J T S; Kern A M, Palermo S, Block B A, Reshotko M R, Chang P L D. Optical I/O Technology for Tera-Scale Computing. *IEEE Journal of solid-state circuits* 2009; 45(1) 235-248.
- [3] Beausoleil R G. Large-Scale Integrated Photonics for High-Performance Interconnects. *ACM Journal on emerging technologies in computing systems* 2011;7(2), Special Issue SIN 6.
- [4] Krishnamoorthy A V, Goossen K W, William J et al. Progress in Low-Power Switched Optical Interconnects. *IEEE Journal of selected topics in quantum electronics* 2011;17(2) 357-376.
- [5] Hoshino K, Yamada K, Matsumoto K, Shimoyama I. Creating a nano-sized light source by electrostatic trapping of nanoparticles in a nanogap. *J. Micromech. Microeng.* 2006;16 1285-1289.
- [6] Nozaki K, Baba T. Laser characteristics with ultimate-small modal volume in photonic crystal slab point-shift nanolasers. *Appl. Phys. Lett.* 2006;88, 211101.
- [7] Nozaki K, Kita S, Baba T. Room temperature continuous wave operation and controlled spontaneous emission in ultrasmall photonic crystal nanolaser. *Optics express.* 2007;15(12) 7506 -7514.

- [8] Egorov V M, Kostsov E G. The prospects for constructing high-performance optical digital computers. *Optoelectronics, Instrumentation and Data Processing*. 1985;1 106-116.
- [9] Egorov V M, Kostsov E G. Microelectronic optical digital computing device. *Optoelectronics, Instrumentation and Data Processing*. 1989;3 61- 68.
- [10] Egorov V. M., Kostsov E. G. Integral optical digital computers. *Appl.Opt.* 1990;29(8) 1178 - 1185.
- [11] Kostsov E. G., Piskunov S.V. 3D ICs with optical interconnections. In: Tverdokhle P.E. (ed.). *3D Laser Information Technologies*. Novosibirsk; 2003. p168-242.
- [12] Markov A. A. *Theory of Algorithms*. Proc. Steklov. Moscow-Leningrad: USSR Academy of Sciences Publishing House. 1954;42.
- [13] Toffoli T., Margolus N. *Cellular automata machines: a new environment for modeling*. Cambridge, Massachusetts: MIT Press; 1987.
- [14] Achasova S., Bandman O., Markova V., Piskunov S. *Parallel Substitution Algorithm. Theory and Application*. Singapore: World Sci.; 1994.
- [15] Ostapkevich M., Piskunov S. The Construction of Simulation Models of Algorithms and Structures with Fine-Grain Parallelism in WinALT. In: V. Malyshkin (ed.) *PaCT 2011*. LNCS. 6873. Heidelberg, Springer; 2011. p192-203.
- [16] WinALT site. <http://winalt.sccc.ru/> (accessed 25 Apr 2012).
- [17] Kartsev M.A., Brick V.A. *Computing systems and synchronous arithmetic*. Moscow: Radio and Communications; 1981.
- [18] Kostsov E.G., Piskunov S.V. Computer design of a two-layer computational array with optical interconnections. *Optoelectronics Instrumentation and Data Processing C/C of Avtometriia*. Allerton Press. 2000;3 3-14.
- [19] Kostsov E G, Piskunov S V, Umrikhina E V. Configurable multilayer pipeline electro-optical structures. *Optoelectronics Instrumentation and Data Processing C/C of Avtometriia*. Allerton Press. 2005;6 62-71.