

Interconnect Analysis of Spatial Decomposition of Boolean Functions for Predictable Nanotechnologies

Svetlana N. Yanushkevich*

University of Calgary, Canada, syanshk@ucalgary.ca,

Bernd Steinbach,

TU Bergakademie Freiberg, Germany, steinb@informatik.tu-freiberg.de,

Vlad P. Shmerko

University of Calgary, Canada, vshmerko@ucalgary.ca

Abstract

This paper addresses the interconnect problem of representation of logic networks in spatial dimensions. Interest to spatial interconnect is motivated by the advent in nanotechnologies and attempts to evaluate and explore the appropriate nanoscale architectures. It has been shown in our previous study that 3D logic network with target topology can be designed by replacing each elementary logic block in a network by its 3D equivalent. In this paper, we solve the problem of partitioning of these 3D blocks over constraints of logic function interconnect and hypercube-like topology. This problem is motivated by the fact that nanotechnologies are sometime critical to the type of this interconnect. We found that decomposition techniques are useful to this solution because they provide possibilities to choose logic functions for assembling decomposed sub-networks.

1 Introduction

The results presented in this paper are relevant to the interconnect problem of a 3D logic network design. Interconnect is critical for upcoming nanotechnologies, in particular, molecular electronics [4, 5, 16], single-electron devices [2], and quantum-dot devices [17], since their high integration may require partitioning into smaller blocks that increase the complexity of interconnect. It has been justified in a number of papers that 3D topologies of computer elements are optimal, physically realistic models for scalable computers [9, 12]. For example, in some molecular technologies, the 3D relationship between molecules is required, in order to achieve the desired functional logical properties [4]. Nanoelectronic devices that are expected to be 3D [5, 16] have motivated efforts to search for new partitioning and decomposition techniques.

Design and implementation of nanodevices is radically different from current technologies. For example, one of possible approaches to design molecular devices is described in [3]. First, a network topology is defined which corresponds to a particular logic function. The electronic structure of the network is described within the topological theory. The active network orbitals are decomposed into atomic orbitals. Their interconnection is described by resonance integrals. The sequence of atomic orbitals and the connections between them are designed to represent wires, network nodes and diodes. In a second step, the electronic properties of the designed network are calculated. The final step is to translate the obtained network model into a molecular chemical structure.

Apart from the traditional gate-level representation translated to transistor-level implementation, another computational structure, Decision Diagrams (DDs) have found their implementation at nanoscale. An example is a solid-state hexagonal nanowire network controlled by Schottky wrap gates [2, 17]. This network is composed of gates demultiplexing an entry into two branches implemented using the designated technology called PADOX. Functionally, this network is a direct mapping of a DD into a nanowire network.

*This work was partially supported by Natural Sciences and Engineering Research Council of Canada (NSERC) and Canadian Foundation for Innovations

In this paper, we study the problem of interconnect in a 3D implementation of a given logic function. In our approach, an interconnect is considered as a connecting media between elementary 3D logic components, called \mathcal{N} -hypercubes. The interconnect is implemented as a special node in the \mathcal{N} -hypercube. The purpose of the interconnect node is to assemble \mathcal{N} -hypercubes while achieving the logical and topological designed characteristics of the 3D network. The inverse task to assembling is partitioning the logic network a given interconnect function. Decomposition techniques provide the systematic approach to solve this problem.

The problem is formulated as follows: given the multi-input gate or a small combinational network, find its 3D partitioning given the topology (\mathcal{N} -hypercube) and the interconnect function. To solve this problem, we chose the decomposition with respect to interconnect function, apply this decomposition to the small network, and then map the decomposed sub-networks into a 3D structure. This mapping of a 2D into a 3D structure preserves the type of interconnect. However, the characteristics of optimality of the decomposed sub-functions in 2D are not necessary transferred, in general, into their 3D representation. That is, the effect of 2D decomposition may not be present in 3D.

We motivate our interest to decomposition techniques it offers a choice of the interconnect functions between the decomposed subparts. Out of our interest are the problems of optimality of 3D decomposed structure. The reason is that the effects of non-optimality are relatively small for decomposition of small functions (elementary logic networks). To the best of our knowledge, this is the first ever attempt to apply the decomposition techniques to 3D hypercube-like structures.

We refer to the previously reported results as follows. Lattice Decision Trees (DTs) and DDs [10] are the result of embedding DTs and DDs into a lattice topology. A 3D decomposition using coordinate representation of Boolean functions have been studied in [1]. Another 3D structure, fat-pyramid [7] provides an opportunity for manipulation of topological properties of multiterminal DTs and DDs. Mapping the DD into the single-electron logic devices has been reported [2]. Topological properties of hypercubes has been discussed in [13]. 3D DD-based logic design methods have been introduced in [18].

In our previous study, the \mathcal{N} -hypercube has been found an appropriate topology to meet the requirements of spatial nanostructures in a 3D logic network design [14, 18, 19, 20]. A 3D logic network is defined as an arbitrary 3D topological structure where the computing elements are the connected \mathcal{N} -hypercubes. In \mathcal{N} -hypercube design, we used the embedding technique in spatial dimensions, in particular [8, 13]. An \mathcal{N} -hypercube is defined as an elementary node of a 3D network. This paper contributes to the interconnect problem of 3D logic network design as follows. First, we utilize the interconnect property of decomposition for extension of interconnect functions between \mathcal{N} -hypercubes. Secondly, we utilized switches for implementing these functions. Both studies satisfy requirements, in particular, of molecular electronics where switches are the most preferable implementation of logic functions.

2 Interconnect of 3D representation of logic networks

2.1 Interconnect of spatial decomposition

Fig. 1 shows various interconnect functions and related decomposition strategies. It follows from this systematization, that the interconnect function can be chosen from the typical library of logic gates. The simplest interconnect function can be implemented on switches or multiplexers. In our approach, a 3D decomposition is defined for small logic networks as embedding of the decomposed functions into \mathcal{N} -hypercubes connected given interconnect function.

2.2 Embedding a logic network into a 3D structure

The embedding means building a guest data structure into an appropriate host structure. Given a computational structure, such as DT, its embedding into a 3D implementation means delegating the computing properties of the DT to the physical structure. The physical structures are offered by technology. In this paper, we consider one of possible 3D structures called \mathcal{N} -hypercube introduced in [14, 18, 19, 20] (Fig. 2a). Note that topology of \mathcal{N} -hypercube is isomorphic to an H -tree [8, 9]. An interconnect function is defined as a logic function to connect two \mathcal{N} -hypercubes.

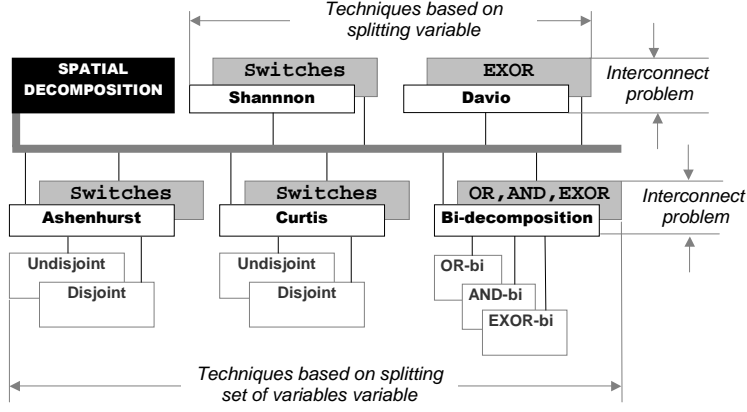


Figure 1: Interconnect problem using state-of-the-art decomposition techniques. Two groups of decomposition methods are distinguished by the splitting strategy: splitting variable or a set of variables.

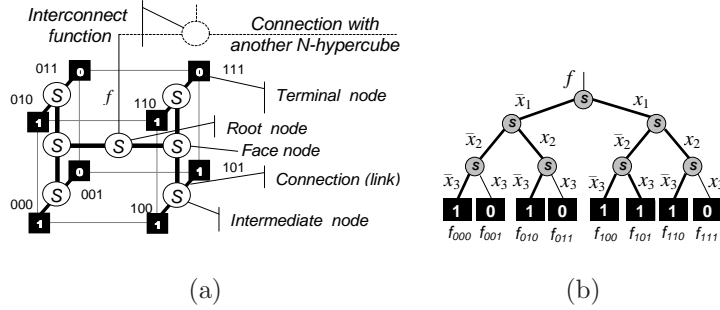


Figure 2: (a) \mathcal{N} -hypercube is an elementary 3D computing element of a 3D logic network. Using technique of embedding, computing properties of decision tree (b) are delegated to the \mathcal{N} -hypercube.

An approach based on embedding DT into \mathcal{N} -hypercube is of exponential complexity. Given a Boolean function on n variables, its complete binary DT, and the corresponding H -tree, are characterized by complexity $O(2^n)$. Hence, it is impossible to embed large Boolean functions into \mathcal{N} -hypercube.

An alternative approach consists in extending the classical technique of multi-level network design by representing each gate by its \mathcal{N} -hypercube equivalent. A targeted 3D topology can be obtained after assembling the \mathcal{N} -hypercubes representing logic gates. Given a network tree of two-input gates g , the network is an incomplete binary tree with g nodes. Hence, embedding in 3D assumes composing of the network of g \mathcal{N} -hypercubes, and the complexity of the proposed approach depends in the number of gates, $O(g)$.

A 3D \mathcal{N} -hypercube is a smallest topological structure in a 3D logic network and represent a Boolean function of three variables (Fig. 2a). Particular cases of an \mathcal{N} -hypercube represent functions of two and one variables. Functions of four and more variables can be represented by connection of several 3D \mathcal{N} -hypercubes. The focus of this paper is the type of interconnect function.

2.3 Interconnect problem at nanoscale

Interconnect is considered with respect to various criteria, for example, number of switches, signal delays, total length, power dissipation, and the cost [7, 8, 12]. Traditional understanding of interconnect as a wire in a voltage-controlled network does not work for certain nanotechnologies [3, 4], especially for those where the carrier of charge (f.e., single electron) is considered as a messenger that travels through the network of nanowires via multiplexing the branches [2]. In such a network, control signals (input variables) are separated from the messenger signal

(function), and the problem of interconnection becomes the problem of compatibility of inputs and outputs. Below we consider switches on multiplexers as the most suitable candidates for interconnect nodes.

3 MUX-based interconnection and decomposition

3.1 Library of interconnect logic elements for spatial decomposition

Interconnect functions are implemented as special node in a 3D network. This node and connected \mathcal{N} -hypercubes can be placed through design in 3D space to achieve the targeted topological characteristics. Interconnect functions must be simple and implemented by switches. In our approach, we used the library of MUX-based implementations (Fig. 3). For example, AND gate can be represented by the MUX $f = 0 \times \bar{s} \vee I_2 \times s = x_2 x_1$. This model is a good candidate for implementation using molecular switches [2] single-electron logic device, composed of nanowires and wrap-gates controlled by voltage signal [3, 6].

The library given in Fig. 3 can be extended to implement three and more input gates. These models correspond to DDs of the gates, and can be implemented by \mathcal{N} -hypercube.

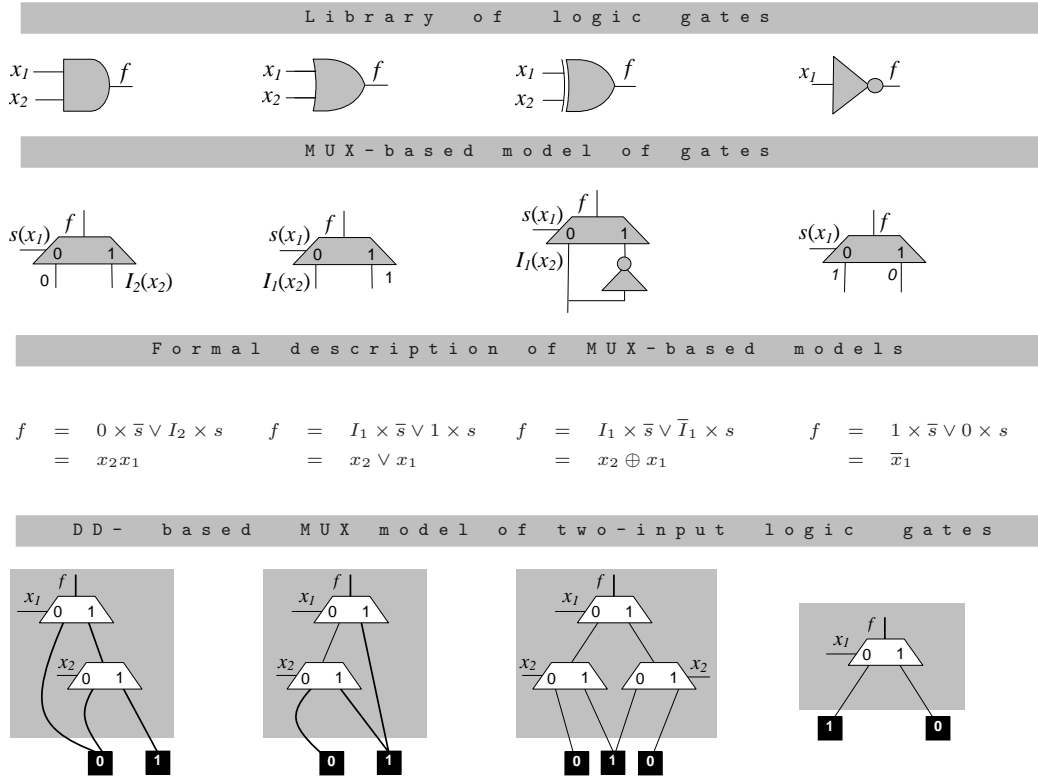


Figure 3: Library of interconnect elements for spatial 3D logic network design includes AND, OR, EXOR, and NOT Boolean functions. These functions can be implemented by switches using multiplexer configuration.

3.2 MUX-based interconnect

The simplest interconnect logic function is the Shannon expansion, S . Davio expansion is used in the form of the *positive* pD expansion $f = f_0 \oplus x_i f_2$, and *negative* Davio nD expansion $f = f_1 \oplus \bar{x}_i f_2$, where $f_0 = f_{x_i=0}$, $f_1 = f_{x_i=1}$, and $f_2 = f_{x_i=0} \oplus f_{x_i=1}$. Both Shannon and Davio expansions can be used in designing 3D networks as interconnect functions. For example, the Boolean function $f = x_1 \bar{x}_2 \vee \bar{x}_3$ is represented by the DT with five active terminal nodes (the value 1) (Fig. 2b). A node in the DT implements the Shannon (S) decomposition $f = \bar{x}_i f_0 \oplus x_i f_1$, where $f_0 = f(x_i = 0)$, and $f_1 = f(x_i = 1)$, for all variables in f . This DT is embedded into

an \mathcal{N} -hypercube (Fig. 2a), where the nodes implement S expansion. Two \mathcal{N} -hypercubes can be connected using the S expansion. Note, that the node function in \mathcal{N} -hypercube corresponds to the chosen Boolean function representation, that is, Shannon, Davio, arithmetic, Walsh, etc. [19].

Let initial an data structure is given by a DD. The problem of spatial decomposition can be formulated in terms of S decomposition as follows: given a DD, find a decomposition in spatial dimension based on \mathcal{N} -hypercube representation. The algorithm for the spatial functional decomposition is as follows:

The algorithm for the MUX-based interconnect using spatial functional decomposition
Step 1: Construct DD
Step 2: Apply decomposition
Step 3: Extend DDs with respect to embedding requirements
Step 4: Embed DDs into \mathcal{N} -hypercubes

This algorithm is illustrated in Fig. 4. A decomposition with respect to the variable x_1 is applied to the root node of the DD that represents a Boolean function f . The decomposed DDs are embedded into \mathcal{N} -hypercubes. Note that DD need some topological corrections before embedding that is easy for small functions. The interconnect is the S expansion implemented by multiplexer.

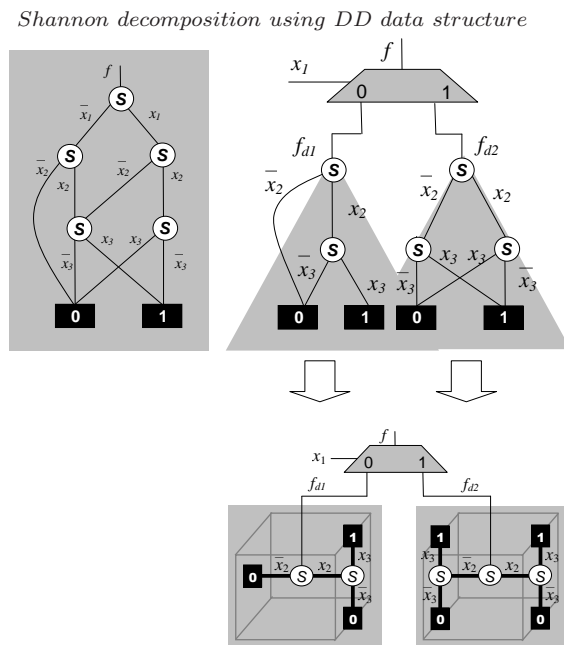


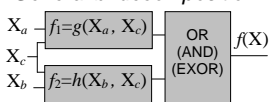
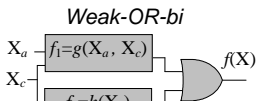
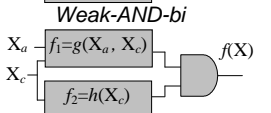
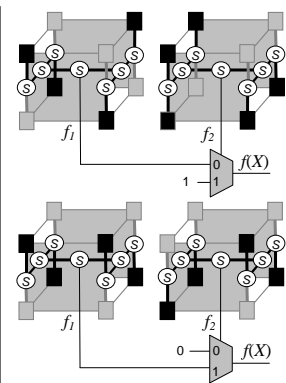
Figure 4: Solution of the spatial interconnect problem using Shannon expansion. Initial DD is decomposed to the DDs of functions $f_{d1}(x_2, x_3) = x_2x_3$ and $f_{d2}(x_2, x_3) = x_2 \oplus x_3$ using multiplexing.

3.3 Interconnect of spatial bi-decomposition

Bi-decomposition [11, 15] provides flexibility in choosing OR, AND, and EXOR interconnect function in 3D network design (Table 1). Disadvantage is that bi-decomposition requires more complicated analysis compared to Shannon and Davio expansions. The algorithm for the spatial bi-decomposition of Boolean function is as follows:

<p>The algorithm for the (AND,OR,EXOR)-based interconnect using bi-decomposition</p> <p>Step 1: Construct DD</p> <p>Step 2: Apply bi-decomposition</p> <p>Step 3: Extend DDs with respect to embedding requirements</p> <p>Step 4: Embed DDs into \mathcal{N}-hypercubes</p> <p>Step 5: Use bi-operation set to joint \mathcal{N}-hypercubes</p>

Table 1: Interconnect functions of a spatial bi-decomposition techniques

DEFINITION	IMPLEMENTATION	3D EXAMPLE
$f(X) = f(X_a, X_b, X_c)$ $= f(g(X_a, X_c), h(X_b, X_c))$ <p>Weak (OR,AND)-bi-decomposition</p> $f(X_a, X_c) = f(g(X_a, X_c), h(X_c))$ <p>AND-bi interconnect: AND OR-bi interconnect: OR EXOR-bi interconnect: EXOR weak-AND-bi interconnect: AND weak-OR-bi interconnect: OR weak-EXOR-bi interconnect: EXOR</p>	<p><i>General bi-decomposition</i></p>  <p><i>Weak-OR-bi</i></p>  <p><i>Weak-AND-bi</i></p> 	

Suppose the OR interconnect function is to be implemented using the \mathcal{N} -hypercubes of elementary logic networks. Suppose that an initial data structure is given by a DD. Fig. 5 shows how bi-decomposition can be implemented using Karplus decomposition: in DD, 0-dominator is a node which belongs to every path from the root to the terminal node 0 and specifies the cut point for OR-bi-decomposition.

OR-interconnect of the decomposed DD

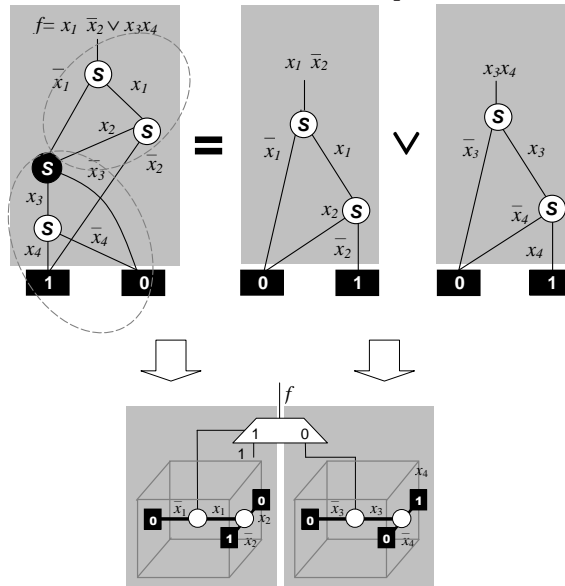


Figure 5: The interconnect for the OR-bi-decomposition: a DD, two decomposed DDs, and corresponding \mathcal{N} -hypercubes connected by a multiplexer.

3.4 Numerical example on interconnect

The goal of experiments was to investigate the feasibility of the proposed approach to the 3D network design. The experiments on MUX-based decomposition of the \mathcal{N} -hypercube models have been conducted on MCNC and ISCAS benchmarks. For each benchmark, a subnetwork with the largest number of gates have been analyzed, and the interconnects between \mathcal{N} -hypercubes have been analyzed for the selected outputs. Note that any preliminary minimization have not been implemented for the benchmark circuits.

In Table 2, the 1st column shows the circuit name **TEST**, the 2nd column provides the number of inputs I , outputs O , and the selected output S with the largest number of gates **I/O(S)**, the 3rd column contains the number of gates G in the circuit and selected subcircuit G_S , **#G(G_S)**. In the next columns, the results of the experiment are given for the S -th subnetwork, and the 4th column contains the number of levels in this subnetwork, **#L**.

Table 2: Fragment of experiments for some middle-size circuits on the interconnect using MUX-based decomposition

2D network representation				3D network		
TEST	I/O(S)	#G(G _S)	#L	#T	#C	#MUX
9sym	9/1(1)	305(305)	8	2086	3872	101
alu2	10/6(2)	730(549)	25	1982	5609	183
alu4	14/8(4)	404(297)	28	3470	10322	99
misex3	14/14(21)	9450(828)	9	6302	12099	275
cordic	23/2(1)	110(70)	15	260	532	22
vg2	25/8(2)	441(96)	7	694	1263	31
c432	36/7(5)	160(126)	17	2022	4319	43
c880	60/26(24)	383(130)	24	612	1585	32
c1908	33/25(25)	880(522)	40	2526	6315	182
c5315	178/123(122)	2307(937)	49	3750	2813	317
c6288	32/32(32)	2416(2327)	124	9246	40050	723

Each gate with more than three inputs or a small logic subnetwork of up to three-input gates have been interpreted as a logic block that requires partitioning into sub-blocks. Each sub-block was represented by an \mathcal{N} -hypercube. The \mathcal{N} -hypercubes and the blocks have been connected using multiplexers. The 5th column of Table 2 contains the number of terminal nodes in the \mathcal{N} -hypercubes, **#T**, the 6th column contains the number of interconnects, **#C**, and the 7th column contains the number of multiplexers, **#MUX**, required for interconnect implementation. For example, the second output is chosen for the circuit alu2. This subnetwork consists of 549 gates in 25 levels. After replacement of each gate in the subnetwork by the \mathcal{N} -hypercube, the number of terminal nodes is equal to 1982, and the number of interconnections is equal to 5609. To connect all \mathcal{N} -hypercubes, 183 multiplexers are needed.

It follows from this experiments, that quite large circuits can be represented using the proposed approach. We anticipate further investigation on application of the decomposition technique for 3D logic circuit design over technological constraints, such as constraints from molecular electronics.

4 Summary

The reported study is focused on the concept of the earlier proposed 3D logic network design. An arbitrary combinational network can be represented in 3D given a target topology. One of technological constraints in 3D network representation is a type of the interconnect function that is used in assembling of \mathcal{N} -hypercubes. We showed that decomposition techniques can be useful in choosing this function for small logic blocks of a network. In our approach, the problem of partitioning in 3D space is replaced by the equivalent problem, that is, decomposition of a network with respect to an interconnect function, and mapping the decomposed subnetworks into 3D. The choice of the interconnect functions depends on the type of decomposition. These functions can be implemented in 3D using the multiplexer-based techniques. The proposed solution of the interconnect based on switches is feasible for large logic networks. Further research is required on application of the proposed approach to design logic networks implemented using

molecular electronics and single-electron technology.

References

- [1] A. Al-Rabady and M. Perkowski, Shannon and Davio sets of new lattice structures for logic synthesis in three-dimensional space, *Proc. 5th Int. Workshop on Applications of the Reed–Muller Expansion in Circuit Design*, Mississippi State Univ., pp.165–184, 2001.
- [2] N. Asahi, M. Akazawa, Y. Amemiya, Single-Electron Logic Device Based on the Binary Decision Diagram, *IEEE Trans. on Electron Devices*, 44(7):1109–1116, 1997.
- [3] D. Crawley, K. Nikolic, M. Forshaw, J. Ackermann, C. Videlot, T.N.Nguyen, L.Wang, P.M. Sarro, 3D Molecular Interconnection Technology, *Journal of Micromechanics and Microengineering*, 13(5):655–662, 2003.
- [4] D. Crawley, K. Nikolić, and M. Forshaw, Eds., *3D Nanoelectronic Computer Architecture and Implementation*, Institute of Physics Publishing, UK, 2005.
- [5] J. C. Ellenbogen and J. C. Love. Architectures for molecular electronic computers: logic structures and an adder designed from molecular electronic diodes. In *Proceedings of the IEEE*, 88:386–426, 2000.
- [6] B. L. Feringa, Ed., *3D Molecular Switches*, Wiley-VCH, 2001.
- [7] R. I. Greenberg, The Fat-Pyramid and Universal Parallel Computation Independent of Wire Delay, *IEEE Trans. Computers*, 43(12):1358–1364, 1994.
- [8] C.H. Leiserson. Fat-trees: universal networks for hardware-efficient supercomputing, *IEEE Trans. Computers*, vol.34, no.10, pp.892–901, 1985.
- [9] C. Mead and M. Rem, Cost and Performance of VLSI Computing Structures, *IEEE J. of Solid State Circuits*, 14(2):455–462, 1979.
- [10] M. A. Perkowski, M. Chrzanowska-Jeske M, and Y. Xu, Lattice Diagrams Using Reed-Muller Logic, *Proc. Int. Workshop Applications of the Reed–Muller Expansion in Circuit Design*, Oxford, UK, pp. 85–102, 1997.
- [11] Ch. Posthoff and B. Steinbach, *Logic Function and Equation – Binary Models for Computer Science*, Springer, 2004.
- [12] A. L. Rosenberg, Three Dimensional VLSI: A Case Study, *J. of ACM*, 30(3):397–416, 1983.
- [13] Öhring, S., and S. K. Das, Incomplete Hypercubes: Embeddings of Tree-Related Networks, *J. Parallel and Distributed Computing*, no. 26, pp. 36–47, 1995.
- [14] V. P. Shmerko and S. N. Yanushkevich, Three-dimensional feedforward neural networks and their realization by nano-devices. *Artificial Intelligence Review, An Int. Science and Engineering Journal* (UK), Special Issue on *Artificial Intelligence in Logic Design*, 20(3-4):473–494, 2004.
- [15] B. Steinbach and Ch. Lang, Exploiting functional properties of Boolean functions for optimal multi-level design by bi-decomposition, In: S. N. Yanushkevich, Ed. *Artificial Intelligence in Logic Design*, pp.159–200, Kluwer, 2004.
- [16] J. M. Tour, W. L. V. Zandt, C. P. Husband, et al., Nanocell Logic Gates for Molecular Computing, *IEEE Trans. on Nanotechnology*, 1(2):100–109, 2002.
- [17] T. Yamada, Y. Kinoshita, S. Kasai, et al., Quantum-dot logic circuits based on shared binary decision diagram. *J. Applied Physics*, 40(7):4485–4488, 2001.
- [18] S. N. Yanushkevich, Logic design of nanodevices, In: M. Rieth and W. Schommers, Eds., *Handbook of Theoretical and Computational Nanotechnology*, Scientific American Publishers, Chapter 110, pp. 1–52, 2005.
- [19] S. N. Yanushkevich, D. M. Miller, V. P. Shmerko, and R. S. Stanković, *Decision Diagram Techniques for Micro- and Nanoelectronic Design*, Taylor & Francis/CRC Press, 2006.
- [20] S. N. Yanushkevich, V. P. Shmerko, and O. R. Boulanov, Embedding and Assembling Techniques for Spatial Computing Structure Design using Decision Trees and Diagrams, *Proc. 36th IEEE Int. Symp. Multivalued Logic*, 2006.