

CONCURRENT TERNARY GALOIS-BASED COMPUTATION USING NANO-APEX MULTIPLEXING NIBS OF REGULAR THREE-DIMENSIONAL NETWORKS, PART III: LAYOUT CONGESTION- FREE EFFECTUATION

Anas N. Al-Rabadi

Department of Computer Engineering, The University of Jordan, Amman – Jordan &
Department of Renewable Energy Engineering, Isra University – Jordan

ABSTRACT

Novel layout realizations for congestion-free three-dimensional lattice networks using the corresponding carbon-based field emission controlled switching is introduced in this article. The developed nano-based implementations are performed in three dimensions to perform the required concurrent computations for which two-dimensional implementations are a special case. The introduced realizations for congestion-free concurrent computations utilize the field-emission controlled switching devices that were presented in the first and second parts of the article for the solution of synthesis congestion and by utilizing field-emission from carbon nanotubes and nanotips. Since the concept of symmetry indices has been related to regular logic design, a more general method called Iterative Symmetry Indices Decomposition that produces regular three-dimensional lattice networks via carbon field-emission multiplexing is presented, where one obtains multi-stage decompositions whenever volume-specific layout constraints have to be satisfied. The introduced congestion-free nano-based lattice computations form new and important paths in regular lattice realizations, where applications include low-power IC design for the control of autonomous robots and for signal processing implementations.

KEYWORDS

Concurrency, Iterative symmetry decomposition, Layout congestion, Lattice networks, Nano-apex emission.

1. INTRODUCTION

Several various methods that are used for functional decomposition are considered as major techniques that are widely used for the synthesis of binary and multi-valued logic circuits and systems [2], [15], [18], [20]-[21], [25]. For example, logic circuits were synthesized using symmetric indices that characterize certain function symmetries like in symmetric networks, Akers arrays and lattice networks [1]-[4], [7], [21]. Other decomposition methods that have been widely utilized to synthesize logic functions utilize function expansions such as in decision trees and diagrams [2], [10], [23], [27]. Other synthesis methods include the use of various synthesis techniques to produce multi-level sum-of-products (SOP) or exclusive-sum-of-products (ESOP) forms for logic circuits and systems [21].

As was previously presented in the first and second parts of the article, regular interconnects generally lead to cheap implementations and high densities, where higher density implies both

higher performance and lower overhead for support components. In addition, the property of high regularity leads to important design consequences such as predictable timing, high testability, fast fault localization and self-repair, and the minimization of power consumption [2], [24], [26]. In this paper, a general processing that produces regular three-dimensional lattice networks via carbon field-emission using the corresponding operations on symmetry indices is presented. A decomposition called the Iterative Symmetry Indices Decomposition (ISID) is implemented for the three-dimensional design of lattice networks, where the synthesis of the special case of regular two-dimensional lattice networks via field-emission techniques using ISID is also presented. The new method can be used for the synthesis of ternary functions using nano-based three-dimensional regular lattice networks whenever volume-specific layout constraints have to be satisfied.

Field electron emission is the emission of electrons from the surface of a cathode under the influence of the applied electric field which is strongly dependent upon the work function of the emitting material [9], [16], [19], where the general form of the governing Fowler-Nordheim equation [17] was produced [16]. Carbon nanotubes (CNTs) are important emerging structures within the highly-expanding science and engineering field of nanotechnology [5], [8], [12], [22], [29]-[30], which have several attractive engineering properties [6], [12]-[14], [22]. Carbon field-emission can be achieved through using various types of single-walled and multi-walled CNTs and also by using carbon nano-apex tips that were presented and utilized in the first and second parts of the article. Figure 1 illustrates the layered layout of the introduced carbon field emission – based system design method that is used in this article.

Three-Dimensional ISID-Based Lattice Realizations
Field Emission-Based Circuits
Carbon-Based Field Emission Devices
Field-Emission Physics
Galois Algebra

Figure 1. The introduced hierarchy for the utilized nano-based system realization.

The research findings and implementations in this article are original and new, and are performed for the first time for the implementation of ternary Galois functions utilizing ISID-based concurrent nano-based lattice systems that use carbon field-emission devices which are based on the field-emission from nano-apex carbon fibers and nanotubes. The operations of the ternary ISID-based three-dimensional lattice architectures utilizing the introduced nano-based systems are also demonstrated.

The remainder of this article is organized as follows: Fundamentals of the important concept of symmetry indices and their application within lattice networks are presented in Section 2. The utilization of ISID-based decomposition for the synthesis of two-dimensional and three-dimensional lattice networks is presented in Section 3. The utilization of the carbon field-emission devices in controlled switching and multiple-valued computing is presented in Section 4. The implementation of controlled switching that use carbon field emission-based devices within ISID-based layout congestion-free lattice architectures via carbon field - emission multiplexing is introduced in Section 5. Conclusions and future work are presented in Section 6.

2. BASICS OF SYMMETRY INDICES

It is well-known in logic synthesis that certain classes of logic functions exhibit specific types of symmetries [2], [11], [21], [28]. Such symmetries have various types that include symmetries

between different functions under negation, symmetries within a logic function under the negation of its variables, and symmetries within a logic function under the permutation of its variables [21]. One method to characterize a symmetry that might exist in a logic function is performed by using symmetry indices.

Definition 1. A single index symmetric function, denoted as $S^k(x_1, x_2, \dots, x_n)$ has value 1 when exactly k of its n inputs are equal to 1, and exactly $(n - k)$ of its remaining inputs are 0.

Example 1. The following Karnaugh map represents symmetric function $F = ab \oplus bc \oplus ac$.

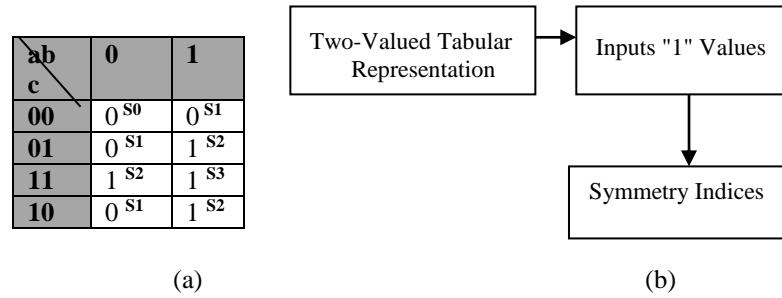


Figure 2. A three-variable symmetric Boolean function: (a) Karnaugh map, and (b) relation with symmetry indices.

In Fig. 2, a symmetry index S^i specifies a Karnaugh map cell that counts value “1” in the specified minterm in number of times equal to i .

Definition 2. The elementary symmetric functions of n variables are:

$$S^0 = \bar{x}_1 \bar{x}_2 \dots \bar{x}_n,$$

$$S^1 = x_1 \bar{x}_2 \dots \bar{x}_n + \bar{x}_1 x_2 \bar{x}_3 \dots \bar{x}_n + \dots + \bar{x}_1 \bar{x}_2 \dots \bar{x}_{n-1} x_n,$$

...

$$S^n = x_1 x_2 \dots x_n.$$

Thus, for Boolean function of three variables one obtains the following sets of symmetry indices: $S^0 = \{\bar{a}\bar{b}\bar{c}\}$, $S^1 = \{\bar{a}\bar{b}c, \bar{a}b\bar{c}, a\bar{b}\bar{c}\}$, $S^2 = \{\bar{a}bc, ab\bar{c}, a\bar{b}c\}$, and $S^3 = \{abc\}$. It has been shown that an arbitrary n -variable symmetric function f is uniquely represented by elementary symmetric functions $\{S^0, S^1, \dots, S^n\}$ as follows: $f = \sum_{i \in A} S^i = S^A$, where $A \in \{0, 1, \dots, n\}$. Also it can be

shown that for $f = S^A$ and $g = S^B$, the following operations are obtained:

$$f \cdot g = S^{A \cap B} \tag{1}$$

$$f + g = S^{A \cup B} \tag{2}$$

$$f \oplus g = S^{A \oplus B} \tag{3}$$

$$\bar{f} = S^{\bar{A}} \tag{4}$$

It has been shown [2], [21] that a function which is not symmetric can be symmetrized by repeating its variables. This method of variable repetition transforms the values of Karnaugh map cells which make the function non-symmetric into don't cares which make the function symmetric.

Example 2. The following Karnaugh map demonstrates the symmetrization by repeating the variables of non-symmetric Boolean function $F = \bar{a} + b$.

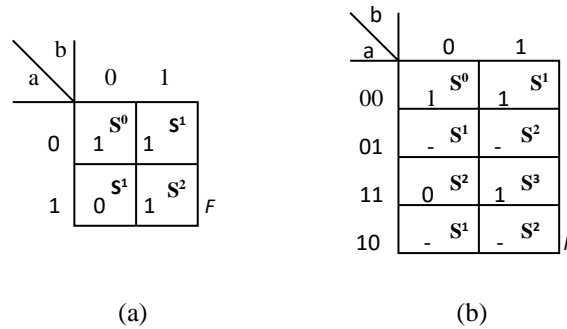


Figure 3. Symmetrization by repeating variables: (a) non-symmetric Boolean function F , and (b) symmetric Boolean function F which is obtained by the repetition of variable $\{a\}$.

One notes that while in Fig. 3(a) conflicting values occur for symmetry index S^1 in minterms $\bar{a}b$ and $a\bar{b}$ and thus producing non-symmetric function, non-conflicting values are produced for the same non-symmetric function in Fig. 3(b) by repeating variable $\{a\}$ two times.

As stated previously, various applications of symmetry indices for the synthesis of logic functions have been previously shown. This includes symmetric networks, Akers arrays and lattice networks [1]-[4], [7], [21] among other several implementations.

The concept of lattice networks for switching functions involves three components: (1) expansion of a function that corresponds to the root in the lattice which creates several successor nodes of the expanded node, (2) joining of several nodes to a single node which is the reverse operation of the expansion process, and (3) regular geometry to which the nodes are mapped. Figure 4(a) shows an example of a four-variable (i.e., four-level) general two-dimensional lattice network (with interconnected nodes which are two-to-one multiplexers), and Figs. 4(b) and 4(c) show the relationship between Fig. 4(a) and symmetry indices.

Therefore, Fig. 4 shows an example of the fact that regular lattice networks exhibit a close relationship with symmetry indices, where symmetry indices represent the sets of all possible paths from leafs to the root of the corresponding lattice network. One also notes that each internal node in Fig. 4(a) is a Shannon node which is practically implemented as a two-to-one multiplexer whose output goes in two directions. Other types of expansion nodes can be implemented as well [2] that will produce corresponding types of various kinds of lattice networks.

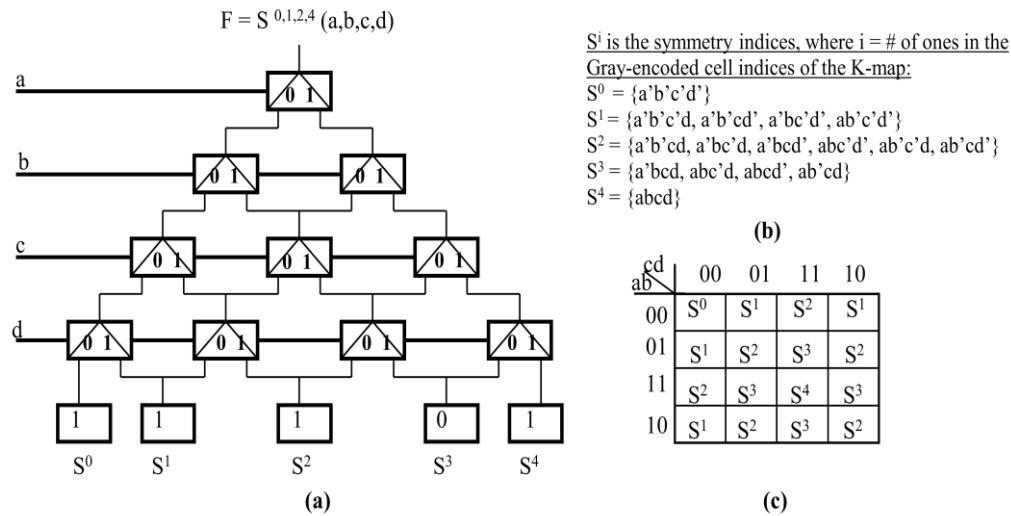


Figure 4. The relation between a two-dimensional lattice and symmetry indices: (a) two-dimensional lattice network for a four-variable function, (b) sets of binary symmetry indices, and (c) the corresponding Karnaugh map interpretation of the binary symmetry indices.

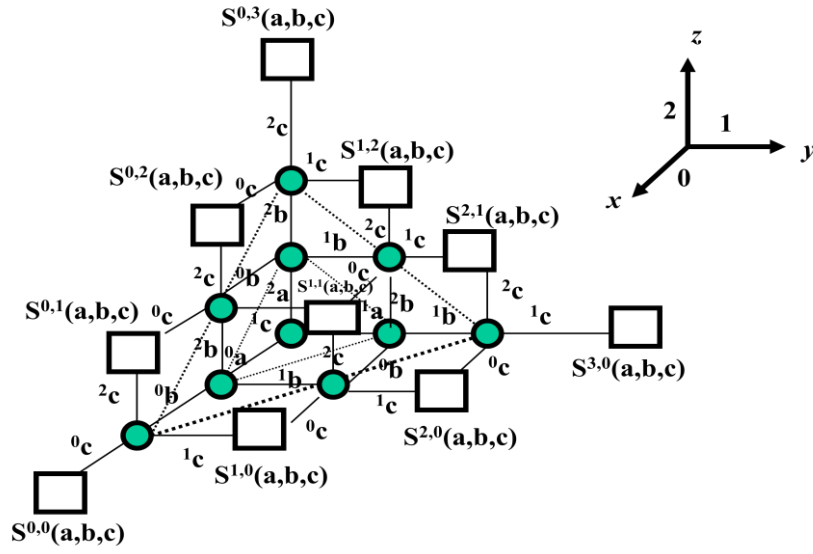
It has been shown [2], [21] that every non-symmetric function can be symmetrized by repeating its variables. Therefore, since a single variable corresponds to a single level in the lattice network, repeating variables produces a repetition of levels in a lattice network. In general, three main factors control the size of a lattice network that realizes non-symmetric functions [2]: (1) expansion types that are used in the internal nodes, (2) order of variables upon which functions are expanded in each level of the lattice, and (3) the choice of repeated variables. Consequently, various optimization methods have been addressed for an optimal choice of the three factors that are mentioned above in order to minimize the size of the corresponding lattice network.

In general, it has been shown [2] that in order to preserve the regular realization of expansions over n^{th} radix, it is sufficient to join n nodes in n -dimensional space to obtain the corresponding lattice networks. For instance, it is sufficient in the binary case to join two nodes. Analogously, it is sufficient in the ternary case to join three nodes to form the corresponding three-dimensional lattice networks. Fully symmetric ternary functions do not need any joining operations to repeat variables in order to realize them in three-dimensional lattice networks.

Figure 5 shows the close relationship between three-dimensional lattice networks and ternary symmetry indices, where ternary symmetry indices are the sets of all possible paths from leafs to the root of a three-dimensional lattice network. Note that each internal node in Fig. 5 is a ternary Shannon node which is a three-to-one multiplexer whose output goes in three directions.

3. LATTICE NETWORK SYNTHESIS USING ITERATIVE SYMMETRY INDICES DECOMPOSITION

The realization of non-symmetric functions using lattice networks demands the repetition of variables [2]. In many cases, one has to repeat variables so many times that will result in a big size two-dimensional lattice network which doesn't fit the specified area or the required volume in the case of three-dimensional networks. On the other hand, one can re-route the corresponding interconnects between the internal nodes of the lattice network using optimization methods in a way such that the network will fit into the specified layout space. Yet, this process will make the used interconnects between lattice cells of many different lengths, and consequently strips the lattice network from one of its most important features for which all of the interconnects are of the same length.



(a)

S^{ij} is the ternary symmetry indices, where i = # of ones in the natural-encoded cell indices of the ternary map, and j = # of twos in the natural-encoded cell indices of the ternary map:

$$\begin{aligned}
 S^{0,0}(a,b,c) &= \{^0a^0b^0c\} \\
 S^{1,0}(a,b,c) &= \{^1a^0b^0c, ^0a^1b^0c, ^0a^0b^1c\} \\
 S^{2,0}(a,b,c) &= \{^1a^1b^0c, ^1a^0b^1c, ^0a^1b^1c\} \\
 S^{3,0}(a,b,c) &= \{^1a^1b^1c\} \\
 S^{0,1}(a,b,c) &= \{^2a^0b^0c, ^0a^2b^0c, ^0a^0b^2c\} \\
 S^{0,2}(a,b,c) &= \{^2a^2b^0c, ^2a^0b^2c, ^0a^2b^2c\} \\
 S^{0,3}(a,b,c) &= \{^2a^2b^2c\} \\
 S^{1,1}(a,b,c) &= \{^1a^2b^0c, ^1a^0b^2c, ^2a^1b^0c, ^2a^0b^1c, ^0a^2b^1c, ^0a^1b^2c\} \\
 S^{1,2}(a,b,c) &= \{^1a^2b^2c, ^2a^1b^2c, ^2a^2b^1c\} \\
 S^{2,1}(a,b,c) &= \{^1a^1b^2c, ^1a^2b^1c, ^2a^1b^1c\}
 \end{aligned}$$

(b)

$ab \backslash c$	0	1	2
00	$S^{0,0}$	$S^{1,0}$	$S^{0,1}$
01	$S^{1,0}$	$S^{2,0}$	$S^{1,1}$
02	$S^{0,1}$	$S^{1,1}$	$S^{0,2}$
10	$S^{1,0}$	$S^{2,0}$	$S^{1,1}$
11	$S^{2,0}$	$S^{3,0}$	$S^{2,1}$
12	$S^{1,1}$	$S^{2,1}$	$S^{1,2}$
20	$S^{0,1}$	$S^{1,1}$	$S^{0,2}$
21	$S^{1,1}$	$S^{2,1}$	$S^{1,2}$
22	$S^{0,2}$	$S^{1,2}$	$S^{0,3}$

(c)

Figure 5. Three-dimensional lattice for ternary three-variable functions: (a) three-dimensional lattice network, (b) sets of ternary symmetry indices, and (c) the ternary natural-encoded map for the ternary symmetry indices.

In current and future circuit technologies, most of the resulting circuit area is occupied by local and global interconnects, and the delay of interconnects is responsible for more than 50% of the total delay which is associated with a circuit or a system [26]. Thus, maintaining local interconnects of equal length will minimize the total length of the used wires and consequently will minimize the resulting delay and consumed power. This idea of maintaining interconnects of equal length for a large size lattice network which does not fit specific layout boundaries can be achieved using the process of Iterative Symmetry Indices Decomposition (ISID) [3].

For a karnaugh map of a non-symmetric Boolean function, conflicting values of “0” and “1” exist within some symmetry indices S^i . While one method of removing such conflicting values is done by repeating variables, another method of removing such conflicting values is done by decomposing the non-symmetric function into a symmetric part superimposed with an error part [3], where the error part can be then iteratively decomposed into a superposition of two parts. The superposition of the decomposed parts to produce the total function can be done using either the Exclusive-OR operator or the Equivalence operator. The following procedure demonstrates the detailed steps for obtaining the required ISID-based decomposition that will maintain interconnects of the same length for large size lattice networks which don't fit specific layout boundaries.

ISID Decomposition	
(1)	For a given area or volume specifications, synthesize any non-symmetric function using a lattice network by repeating variables.
(2)	If repeating variables will force the lattice network to grow out of the layout boundaries, decompose the non-symmetric function into two superimposed parts; a symmetric part and an error part (i.e., the correction part). Then, the original function is equal to the exclusive-OR or the equivalence of the two decomposed functions. This is denoted as \oplus -ISID and \otimes -ISID, respectively. Since there are many possible ways to obtain a symmetric function from the original non-symmetric function using ISID, one can choose a symmetric part by using the optimization criterion of Hamming distance in which one chooses the minimum number of changes of function values that are needed to transform the non-symmetric Boolean function into a symmetric one.
(3)	Synthesize the symmetric part using a lattice network. If the synthesis fits layout boundaries then synthesize the error function.
(4)	If the resulting synthesis does not fit layout boundaries, then go to step (2) and perform in serial-mode a single decomposition of the symmetric or error sub-functions, or perform in parallel-mode a multi-decomposition on all symmetric and error sub-functions.
(5)	Repeat step (4) until the synthesis fits the specified layout boundaries.

Example 3. For the non-symmetric function F , Fig. 6 shows the synthesis of F using a two-dimensional lattice network which requires the repetition of a variable $\{b\}$ to remove the non-symmetrization.

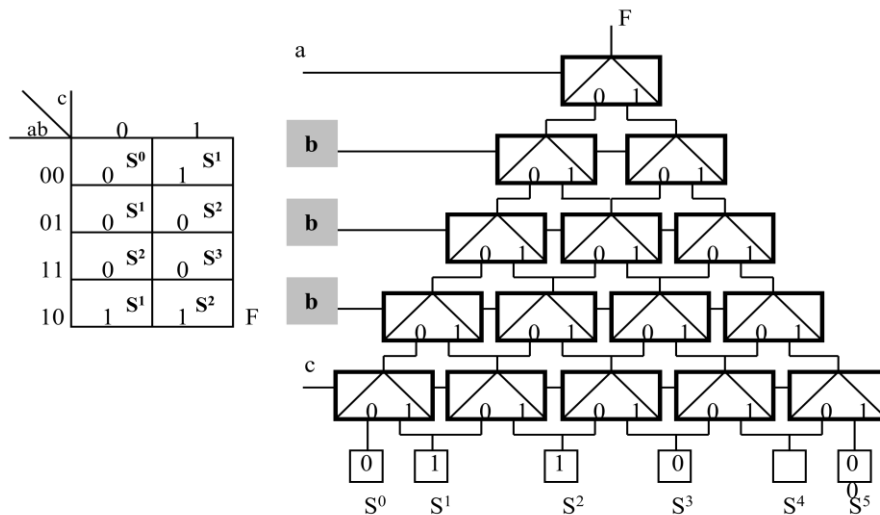


Figure 6. A two-dimensional Shannon lattice network.

One notes that the two-dimensional lattice network in Fig. 6 is made up of 15 two-to-one multiplexers. Each two-to-one multiplexer consists of three logic gates. Thus, Fig. 6 is made up of a total of 45 logic primitives. Using the ISID procedure, the same non-symmetric function can be synthesized using XOR-based ISID as shown in Fig. 7.

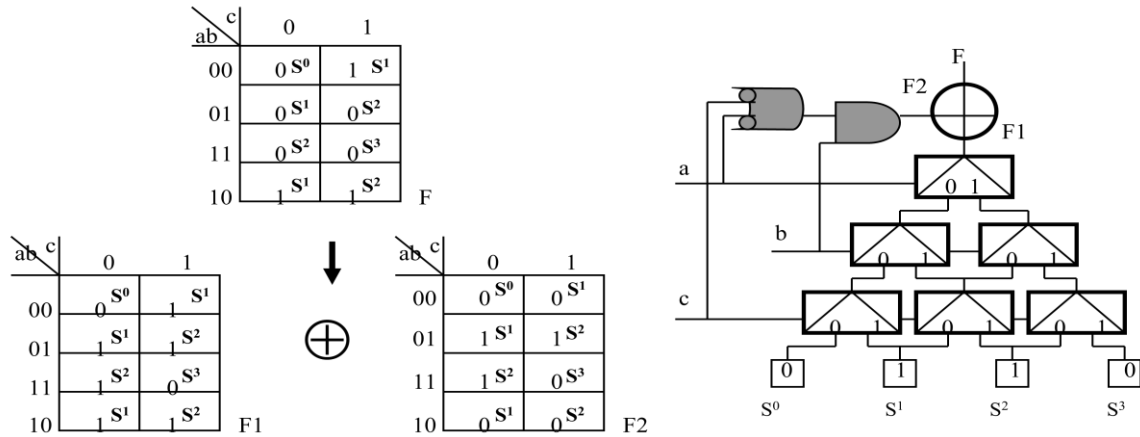


Figure 7. Shannon ISID XOR-based lattice network.

In Fig. 7, $F1$ realizes the symmetric part and $F2$ realizes the error part. Note that in Fig. 7 one has six two-to-one multiplexers and three logic primitives namely AND, OR and XOR. Thus, one has a total of $6 \cdot 3 + 3 = 21$ logic primitives. Consequently, by comparing the total number of gates needed in Fig. 7 to those in Fig. 6, one observes that we economized a total of 24 logic primitives. On the other hand, if one uses ISID XNOR-based synthesis then we obtain the decomposition in the tables that are shown in Fig. 8 and the corresponding lattice-based synthesis of the non-symmetric function. The cost of the lattice network in Fig. 8 is 21 logic primitives which is equal to the cost that was obtained in Fig. 7.

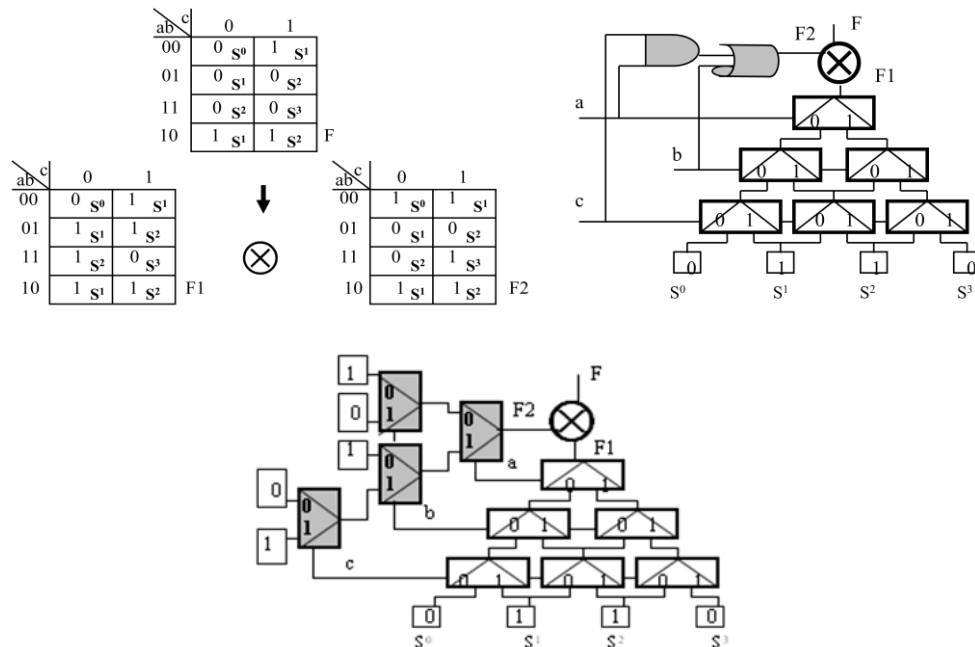


Figure 8. Shannon ISID XNOR-based lattice network.

One observes that two-level sum-of-product (SOP) circuits have been used to synthesize the error functions in Figs. 7 and 8, respectively. This choice has the advantage of the use of minimal number of logic primitives, but has the disadvantage of transforming the lattice network from a fully-regular network, where only one type of primitives has been used (namely the two-to-one multiplexer), to a semi-regular network where many different logic primitives have been used; two-to-one multiplexer, AND and OR gates. To maintain full regularity in terms of using one type of logic primitives, one can synthesize the error function using a separate multiplexer-based logic network. This idea is demonstrated in Fig. 8 using the ISID XNOR-based Shannon lattice network, where one observes that the multiplexer-based network in Fig. 8 has 10 two-to-one multiplexers and thus has 30 Boolean gates, which is still much less than the total number of gates that were obtained in Fig. 6.

Using the procedure for the ISID decomposition, one can have a complicated decomposed structure that fits certain specifications. The following extends the ISID algorithm to the case of the three-dimensional space, where the introduced three-dimensional logic synthesis is implemented using the ternary radix of Galois field which is shown in Table 1.

Table 1. Third radix Galois operations: (a) GF(3) addition and (b) GF(3) multiplication.

(a)				(b)			
+	0	1	2	*	0	1	2
0	0	1	2	0	0	0	0
1	1	2	0	1	0	1	2
2	2	0	1	2	0	2	1

Analogously to the ISID procedure for the binary case of two-dimensional lattice network, the three-dimensional lattice network of ternary functions using ISID can be obtained as well. The same idea of two-dimensional ISID can be used for three-dimensional ISID by using the corresponding algebraic equations over GF(3) to decompose the corresponding three-valued input three-valued output maps, where these equations are shown as follows:

$$a *_3 a *_3 a = a \quad (5)$$

$$a +_3 a +_3 a = 0 \quad (6)$$

Thus, the same ISID procedure (which was previously presented) is utilized in three-dimensions, with the only exception of replacing binary Galois operations (e.g., the XOR operation) with the corresponding ternary Galois operations that are represented in Equations (5) and (6). Full illustration of this will be presented in Section 5 where the synthesis of nano-based ISID three-dimensional lattice networks will be shown.

4. MULTIPLE-VALUED GALOIS COMPUTING USING CARBON FIELD EMISSION – BASED DEVICES

As was previously shown in the first and second parts of the article, and by utilizing the previously experimented and observed characterizations and operations of carbon field-emission that were presented, Fig. 9 presents the carbon field-emission primitive that realizes the two-to-

one controlled switch. In Fig. 9, the input control signal that is used to control the electric conduct of the device is implemented using the imposed electric field intensity (E) or equivalently the corresponding work function (Φ) or voltage (V).

In the carbon field-emission device which is shown in Fig. 9(a), by imposing the control signal of high voltage (HV), the voltage difference between the carbon cathodes and the facing anode is varied. This change will make the carbon cathode with control signal (HV) to be field-emitting while the other carbon cathode with the complementary control signal ($\overline{\text{HV}}$) to be without field-emission. When the voltage difference is reversed, then carbon cathode with the complementary control signal ($\overline{\text{HV}}$) will be field-emitting while the other carbon cathode with the control signal (HV) will be without field-emission. Thus, this device implements the functionality of the two-to-one controlled switching ($G = ac + bc'$) which is shown in Fig. 9(b).

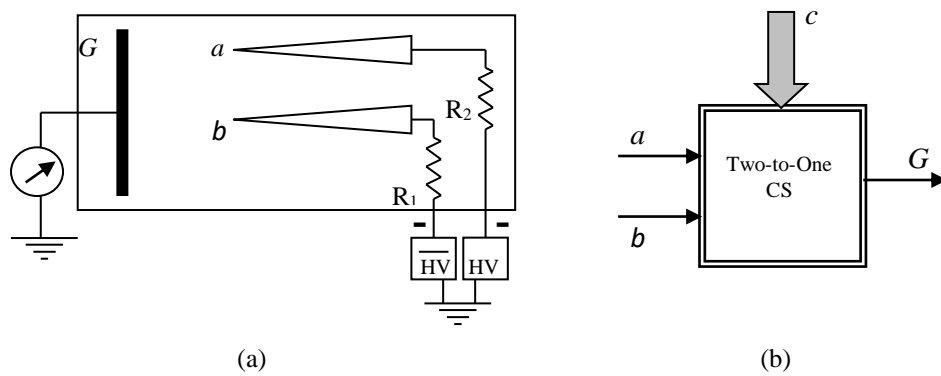


Figure 9. Carbon field-emission device implementing two-to-one controlled switching (CS): (a) carbon field-emission two-to-one controlled switching, and (b) the corresponding block diagram.

Synthesizing many-to-one carbon field-emission controlled switches is possible using the fundamental two-to-one carbon field-emission controlled switch from Fig. 9. For example, for the three-valued logic case, one needs two devices from Fig. 9 to realize the functionality of three-to-one carbon field-emission controlled switch. In general, for the general case of m -valued logic, one needs $(m-1)$ of the two-to-one controlled switches to realize the function of an m -to-1 controlled switching. This idea is illustrated in Fig. 10 where devices $\{D_1, \dots, D_{(m-1)}\}$ can be realized using the carbon-based field-emission controlled switch that was presented in Fig. 9.

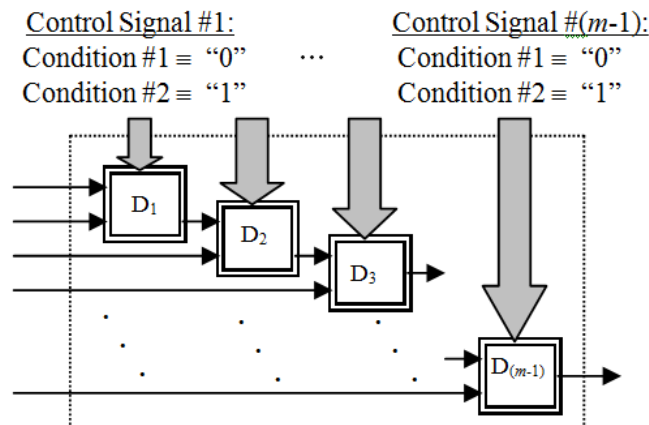


Figure 10. The realization of a general (m -to-1) controlled switch.

Multiple-valued computing has been also illustrated using the presented carbon-based field-emission device. A controlled switch-based circuit that implements GF(3) addition and multiplication tables is shown in Fig. 11, where Fig. 11(a) can be implemented using the two-input single-output carbon field-emission device from Fig. 9.

In Fig. 11(b), variables $\{A, B\}$ are two ternary input variables that can take any value from the set $\{0, 1, 2\}$, inputs $\{0, 1, 2\}$ are constant inputs, and inputs C_k ($k = 0, 1, 2, 3$) are two-valued control variables that take values from the set $\{0, 1\}$. Note that Fig. 11(b) implements GF(3) addition and multiplication tables by using the appropriate values of control variables C_k that select the corresponding device inputs of variable inputs $\{A, B\}$ and constant inputs $\{0, 1, 2\}$.

For instance, Table 2 shows an example for the implementation of GF(3) addition and multiplication operations using Fig. 11(b). In Table 2, the symbol (+) means GF(3) addition, symbol (*) means GF(3) multiplication, $C_k (+)$ means that the control variable C_k to implement the ternary addition operation, and $C_k (*)$ means that the control variable C_k to implement the ternary multiplication operation.

Since three-valued networks over GF(3) will be synthesized using the corresponding addition and multiplication operations, the circuit in Fig. 11(b) can be used in multi-valued implementations whenever GF(3) addition and multiplication operations are applied. For example, any hierarchical complex system-level implementation can be performed from the iterative utilization of the carbon-based controlled switching device which is shown in Fig. 11, where the corresponding arithmetic operations can be implemented using the nano circuit from Fig. 11(b) and by utilizing the corresponding specified input values from Table 2.

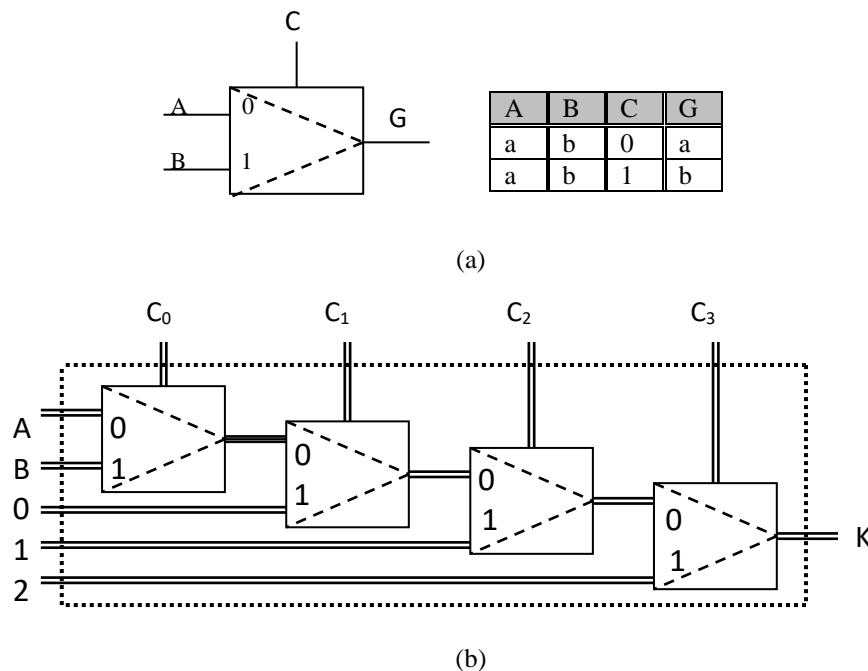


Figure 11. The carbon-based implementation of Galois arithmetic operations: (a) controlled switch symbol that can be implemented using the device in Fig. 9, and (b) circuit that uses controlled-switching to implement GF(3) addition and multiplication operations.

Table 2. The implementation of third radix Galois addition and multiplication operations using Fig. 11(b).

A	B	C ₀ (+)	C ₁ (+)	C ₂ (+)	C ₃ (+)	C ₀ (*)	C ₁ (*)	C ₂ (*)	C ₃ (*)	+	*
0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0
0	2	1	0	0	0	0	0	0	0	2	0
1	0	0	0	0	0	1	0	0	0	1	0
1	1	0	0	0	1	0	0	0	0	2	1
1	2	0	1	0	0	1	0	0	0	0	2
2	0	0	0	0	0	1	0	0	0	2	0
2	1	0	1	0	0	0	0	0	0	0	2
2	2	0	0	1	0	0	0	1	0	1	1

5. LAYOUT CONGESTION-FREE LATTICE PROCESSING VIA CARBON FIELD - EMISSION MULTIPLEXING

This section introduces the synthesis of three-valued Galois functions using carbon field emission-based three-dimensional lattice networks utilizing the method shown in Fig. 12. In this method, mapping any three-valued function into the corresponding three-dimensional lattice network can be achieved using either of the following two functional forms: (1) the function expression form through the RPL-based decomposition, or (2) using the tabular form of the corresponding three-valued function.

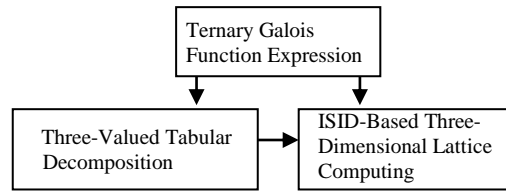


Figure 12. Utilized method to realize three-valued Galois logic using function decompositions.

Example 4 shows the realization of a ternary non-symmetric function in a three-dimensional lattice through the repetition of variables and by utilizing the synthesis scheme from Fig. 12, where in general the operations performed in each node in the three-dimensional lattice network can be implemented using the nano-based circuit from Fig. 11(b) and by using the specified input values from Table 2.

Example 4. Lets us apply the ternary version of ISID procedure to decompose the map in Fig. 13(a).

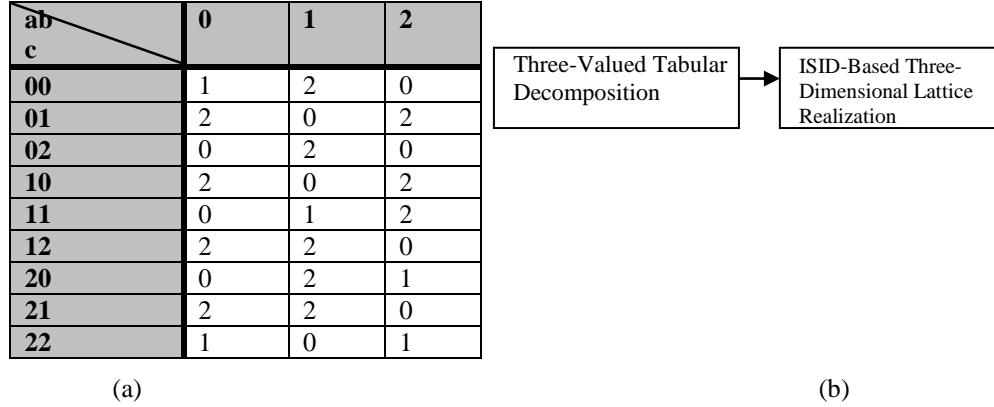


Figure 13. A non-symmetric ternary function: (a) Ternary map, and (b) ISID to be performed using the tabular decomposition of the ternary function.

Using the previously presented definition of ternary symmetric indices, one can observe that the ternary non-symmetric map shown in Fig. 13 can be decomposed using the ISID procedure into a symmetric part and non-symmetric part. In contrast to the binary case, where this can be done only in one specific EXOR expression, ISID can be done in the multiple-valued case in various ways. This can be illustrated here as the indexed cell $\{a = 0, b = 2, c = 2\}$ has the value of “0”. To produce a symmetric part, the indicated indexed cell has to have the value of “1”. Consequently, there are two possibilities for this. The first one follows Equation (6) for which $(1 +_{GF(3)} 1 +_{GF(3)} 1 = 0)$, and the second possibility follows the algebraic equation from Table 1(a) for which $(1 +_{GF(3)} 2 = 0)$. Thus two possible ternary map decompositions follow as shown in Fig. 14, and consequently two possible three-dimensional lattice networks can be implemented using the ternary ISID decomposition as shown in Fig. 15. One notes that, for the non-symmetric function in Fig. 13, one needs to repeat one of the variables in order to realize such non-symmetric ternary function using a three-dimensional lattice network.

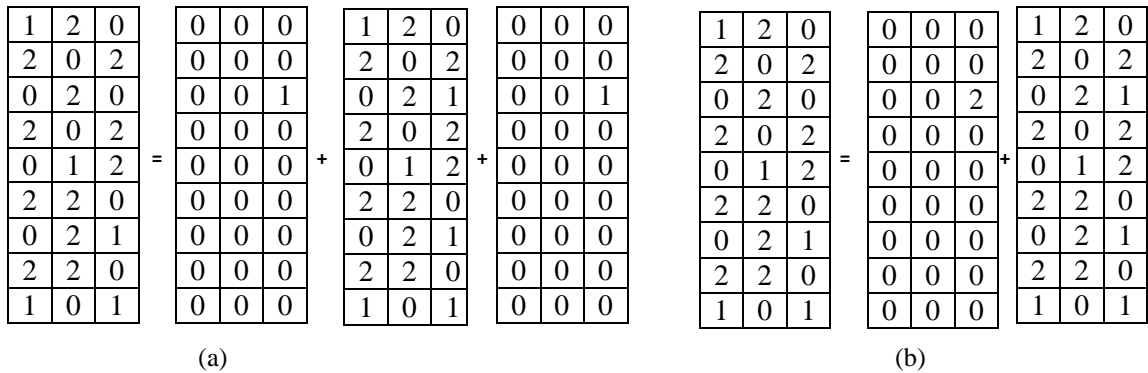


Fig. 14. Two possible ISID decompositions for the ternary three-variable $\{a, b, c\}$ function in Fig. 13: (a) first decomposition obtained using the GF(3) algebraic equation $(1 +_3 1 +_3 1 = 0)$, and (b) second decomposition obtained using the GF(3) algebraic equation $(1 +_3 2 = 0)$.

The process of variable repetition in a three-dimensional lattice network will impose the addition of many new internal cells, depending on the variables chosen to be repeated, and in the worse case one has to repeat variables so many times that the final three-dimensional lattice network

will not fit the specified three-dimensional volume boundaries. On the other hand, the networks in Fig. 15 do not need to repeat variables because the ternary error part is realized in Galois field Sum-Of-Product (GFSOP) network. This can reduce substantially the number of nodes that are needed for highly non-symmetric ternary functions. Yet, the trade off here is that, while we have a totally-regular structure in the case of large three-dimensional lattice networks where one uses only single type of internal nodes and only one type of interconnects, we have now (as shown in Fig. 15) a semi-regular lattice network which is composed of two parts, namely the error part which is not fully-regular in general, and the symmetric part which is made of a fully-regular three-dimensional lattice network.

Analogously to the two-dimensional case, three-dimensional network synthesis using ternary ISID can be iteratively performed in either serial-mode or parallel-mode. The result of such three-dimensional ISID will be the decomposition of a large three-dimensional lattice network into many superimposed smaller three-dimensional lattice networks as shown in Fig. 16; the iterative use of ISID (in serial or parallel modes) will result in decomposing the three-dimensional cubical lattice into corresponding three-dimensional pyramids. The resulting decomposition seen in Fig. 16 will generate pyramids in three dimensions, in contrast to the binary case for which the iterative use of ISID will decompose the two-dimensional rectangular layout into many smaller two-dimensional triangles.

One also can note that the error part in a ternary ISID can be realized in a three-dimensional lattice network with variable repetition if the error function is non-symmetric, or without any variable repetition if the error function is symmetric. Thus, one can maintain full regularity in the total interconnected network that includes both the error part and the symmetric part, since one will use only one type of internal nodes and only one type of internal interconnects.

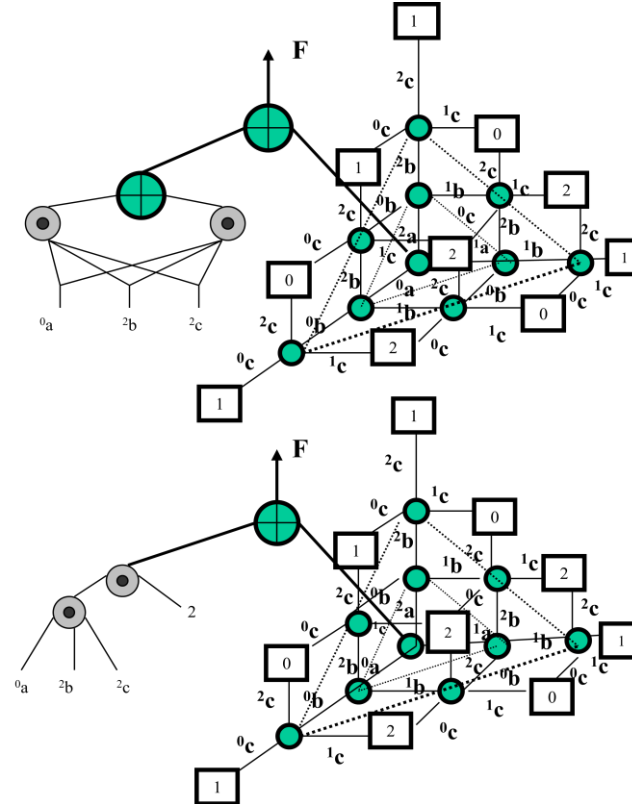


Figure 15. Two functionally-equivalent realizations of three-dimensional lattice networks using the corresponding implementations of ternary ISID from Figs. 14(a) and 14(b), respectively.

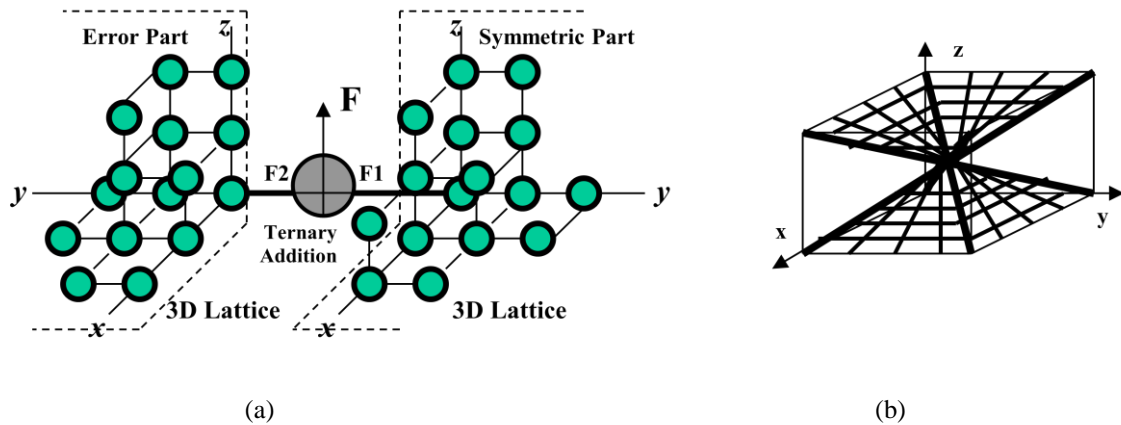


Figure 16. The iterative implementation of ISID procedure for the decomposition of ternary non-symmetric functions in three-dimensions: (a) ISID in three-dimensional lattice networks, and (b) pyramid grid layout as a result of the iterative implementation of the ISID procedure.

As mentioned previously, for the corresponding ternary three-dimensional lattice networks (such as the ones shown in Fig. 15), the operations performed in each internal node can be implemented using the nano circuit from Fig. 11(b) and by utilizing the corresponding specified input values from Table 2, from which all internal node operations over GF(3) can be achieved.

The synthesized resulting ISID-based three-dimensional lattice networks (such as the ones illustrated in Fig. 15) still possess the important characteristic of regularity, since the error part in a ternary ISID can be realized in a three-dimensional lattice network with variable repetition if the error function is non-symmetric or without any variable repetition if the error function is symmetric. Therefore, one can maintain full regularity in the total interconnected network that includes both of the error and symmetric parts, and thus preserving the important properties of compactness in three-dimensional space, ease of manufacturability, no need for three-dimensional layout routing and placement, predictable timing, relative ease of testability and repair when fault occurs, and lower power consumption which is due to using both (a) only local interconnects and (b) low-power carbon nano-based switches that were demonstrated in Fig. 11.

6. CONCLUSIONS AND FUTURE WORK

The synthesis of three-dimensional lattice networks using carbon field-emission nano switching devices and Iterative Symmetry Indices Decomposition (ISID) for congestion-free low-power layout realization is introduced. The operations of the ternary ISID-based three-dimensional lattice architectures utilizing the introduced nano-based systems are also demonstrated. The application of ISID-based decomposition in three dimensions uses operations on symmetry indices that superimpose iteratively a symmetric part and an error-part of an arbitrary non-symmetric function, which allows for achieving congestion-free nano-based realizations of three-dimensional lattice networks, within which internal nodes are synthesized using carbon field-emission nano multiplexing devices.

The research realizations which are introduced in this article are new and original, and are performed for the first time to implement ternary Galois functions utilizing concurrent ISID-based nano lattice systems that use carbon field-emission devices which are based on the field-emission from carbon nano-apex fibers and nanotubes.

The new implementation of congestion-free architectures are important since the resulting nano-based regular three-dimensional lattice networks possess the highly important properties of regularity and low- power consumption, which are required in several synthesis applications. Due to the use of local regular interconnects and low-power carbon-based nano switching devices, the resulting three-dimensional lattice networks can be especially important for future three-dimensional technologies for which major requirements include the minimization of power consumption and performance enhancement.

Future work will include items such as: (1) investigating the use of various optimization algorithms to obtain minimal-size three-dimensional lattice networks by utilizing optimization methods such as (a) optimal selection of the order of control variables and (b) optimal selection of the utilized types of internal lattice nodes, (2) further evaluations of the introduced methods using applied metrics such as size, delay and power consumption in order to quantify the efficiency of the presented new realizations, and (3) the complete fabrication and testing of the new carbon field-emission controlled-switching devices and their integrated application within system-level nano-based lattice architectures for arithmetic-intensive computing applications.

ACKNOWLEDGEMENT

This research was performed during sabbatical leave in 2019-2020 granted to the author from The University of Jordan and spent at Isra University, Jordan.

REFERENCES

- [1] S. B. Akers, "A rectangular logic array," IEEE Transactions on Computers, Vol. C-21, pp. 848-857, 1972.
- [2] A. N. Al-Rabadi, *Reversible Logic Synthesis: From Fundamentals to Quantum Computing*, Springer-Verlag, New York, 2004.
- [3] A. N. Al-Rabadi, "Three-dimensional lattice logic circuits, part III: solving 3D volume congestion problem," Facta Universitatis – Electronics and Energetics, Vol. 18, No. 1, pp. 29 – 43, 2005.
- [4] A. N. Al-Rabadi and M. Perkowski, "New families of reversible expansions and their regular lattice circuits," Journal of Multiple-Valued Logic and Soft Computing, Old City Publishing, U.S.A., Vol. 11, No. 3-4, pp. 213 – 238, 2005.
- [5] A. N. Al-Rabadi, "New dimensions in non-classical neural computing, Part I: three-dimensionality, invertibility, and reversibility," International Journal of Intelligent Computing and Cybernetics, Emerald, Vol. 2, No. 2, pp. 348-385, 2009.
- [6] A. N. Al-Rabadi, *Carbon NanoTube (CNT) Multiplexers, Circuits, and Actuators*, United States Patent and Trademark Office, Patent No. US 7,508,039 B2, U.S.A., 2009.
- [7] A. N. Al-Rabadi, "Circuits for m-valued classical, reversible and quantum optical computing with application to regular logic design," Int. Journal of Intelligent Computing and Cybernetics, Emerald, United Kingdom, Vol. 2, No. 1, pp. 52 – 101, 2009.
- [8] A. N. Al-Rabadi, "New dimensions in non-classical neural computing, Part II: quantum, nano, and optical," International Journal of Intelligent Computing and Cybernetics, Emerald, Vol. 2, No. 3, pp. 513-573, 2009.
- [9] J. M. Bonard, J. P. Salvetat, T. Stöckli, L. Forro, and A. Chatelain, "Field emission from carbon nanotubes: perspectives for applications and clues to the emission mechanism," Applied Physics A: Materials Science & Processing, Vol. 69, No. 3, pp. 245-254, 1999.
- [10] R. E. Bryant, "Graph-based algorithms for Boolean functions manipulation," IEEE Transactions on Computers, Vol. C-35, No. 8, pp. 667-691, 1986.
- [11] J. T. Butler and K. A. Schueller, "Worst case number of terms in symmetric multiple-valued functions," Proceedings of the International Symposium on Multiple-Valued Logic, pp. 94-101, Victoria, B. C., Canada, May 26- 29, 1991.
- [12] P. G. Collins and P. Avouris, "Nanotubes for electronics," Scientific American, pp. 62-69, 2000.
- [13] P. G. Collins, M. S. Arnold, and P. Avouris, "Engineering carbon nanotubes and nanotube circuits using electrical breakdown," Science, Vol. 292, 2001.

- [14] V. Derycke, R. Martel, J. Appenzeller, and P. Avouris, "Carbon nanotube inter- and intramolecular logic gates," *Nano Letters*, Vol. 0, No. 0, A – D, 2001.
- [15] B. Falkowski and S. Rahardja, "Classification and properties of fast linearly independent logic transformations," *IEEE Transactions on Circuits and Systems-II*, Vol. 44, No. 8, pp. 646-655, 1997.
- [16] R. G. Forbes, "Extraction of emission parameters for large-area field emitters, using a technically complete Fowler–Nordheim-type equation," *Nanotechnology*, IOP Publishing, 23(9), 2012.
- [17] R. H. Fowler and L. W. Nordheim, "Electron emission in intense electric fields," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 119(781):137-181, 1928.
- [18] D. H. Green, "Families of Reed-Muller canonical forms," *Int. Journal of Electronics*, No. 2, pp. 259-280, 1991.
- [19] P. Hommelhoff, Y. Sortais, A. Aghajani-Talesh, and M. A. Kasevich, "Field emission tip as a nanometer source of free electron femtosecond pulses," *Physical Review Letters*, 96(7):077401-1:4, 2006.
- [20] M. G. Karpovsky, *Finite Orthogonal Series in the Design of Digital Devices*, Wiley, 1976.
- [21] Z. Kohavi, *Switching and Finite Automata Theory*, McGraw-Hill, New York, 1978.
- [22] K. Likharev, "Electronics below 10 nm," in: *Nano and Giga Challenges in Microelectronics*, pp. 27-68, Elsevier, 2003.
- [23] M. Radmanovic and R. S. Stankovic, "Generating synthetic MVL benchmarks from random MDDs under restrictions," *IEEE Int. Symposium on Multiple-Valued Logic*, Vol. 1, pp. 168-173, 2018.
- [24] S. M. Reddy, "Easily testable realizations of logic functions," *IEEE Trans. on Computers*, C-21, pp. 1183-1188, 1972.
- [25] K. Roy and S. Prasad, *Low-Power CMOS VLSI Circuit Design*, John Wiley & Sons Inc, 2000.
- [26] K. C. Smith, "Prospects for VLSI technologies in MVL," Booklet of ULSI Workshop, p. 4, Boston, Massachusetts, 2002.
- [27] R. Stankovic, "Functional decision diagrams for multiple-valued functions," *Proc. Int. Symp. on Multiple-Valued Logic*, pp. 184-189, 1995.
- [28] R. Tosic, I. Stojmenovic, and M. Miyakawa, "On the maximum size of the terms in the realization of symmetric functions," *Proc. of the International Symposium on Multiple-Valued Logic*, pp. 110-117, Victoria, B.C., Canada, May 26-29, 1991.
- [29] L. Zhang, S. Fahad, H.-R. Wu, T.-T. Dong, Z.-Z. Chen, Z.-Q. Zhang, R.-T. Liu, X.-P. Zhai, X.-Y. Li, X. Fei, Q.-W. Song, Z.-J. Wang, L.-C. Chen, C.-L. Sun, Y. Peng, Q. Wang, and H.-L. Zhang, "Tunable nonlinear optical responses and carrier dynamics of two-dimensional antimonene nanosheets," *Nanoscale Horizons*, 5(10), pp. 1420-1429, 2020.
- [30] Z. Zhu, J. Guan, and D. Tomanek, "Strain-induced metal-semiconductor transition in monolayers and bilayers of gray arsenic: A computational study," *Physical Review B*, 91, 161404(R), 2015.

AUTHOR

Anas N. Al-Rabadi is currently a Professor in the Department of Computer Engineering at The University of Jordan. He received his Ph.D. in Computer Design and Advanced Logic Synthesis in 2002 and M.Sc. in Control and Power Electronic Systems Design in 1998, both from the Department of Electrical and Computer Engineering at Portland State University. From 2002 until 2004, Prof. Al-Rabadi had served as a Research Faculty at the Office of Graduate Studies and Research at Portland State University. He is the author of the first international published title and first comprehensive book on Reversible Logic Synthesis (Springer-Verlag, 2004), *Reversible Logic Synthesis: From Fundamentals to Quantum Computing*. Currently, Prof. Al-Rabadi is the author of more than 130 international scholarly and indexed publications. He also holds a USPTO nanotechnology patent which is registered in 2009 in U.S.A. under patent No. US 7,508,039 B2. His current research interests include distributed and parallel computing, systolic architectures, regular circuits and systems, reversible logic, quantum computing, multiple-valued logic, soft computing and computational intelligence, machine learning, artificial intelligence, optical computing, reconstructability analysis of systems, spectral methods, signal processing, testing and design for testability, nanotechnology, robotics, optimal and robust control, and digital error-control coding.