

Novel Joint Cybersecurity-Channel Coding Scheme via Homeomorphism and Nano Controlled-Switching Lattice Grids, Part II: Regular Hierarchical Realization

Anas N. Al-Rabadi^{1*}, Rania F. Rabadi²

¹Department of Computer Engineering, School of Engineering, The University of Jordan, Amman, Jordan

²German Association for International Cooperation (GIZ), Amman, Jordan

Email: *alrabadi@yahoo.com, a.alrabadi@ju.edu.jo, raniafalrabadi@yahoo.com

How to cite this paper: Al-Rabadi, A.N. and Rabadi, R.F. (2026) Novel Joint Cybersecurity-Channel Coding Scheme via Homeomorphism and Nano Controlled-Switching Lattice Grids, Part II: Regular Hierarchical Realization. *Journal of Computer and Communications*, 14, 115-154. <https://doi.org/10.4236/jcc.2026.149008>

Received: July 28, 2026

Accepted: September 27, 2026

Published: September 30, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Regular hierarchical architecture design using lattice grids via nano-based carbon controlled-switching devices is implemented for the newly introduced homeomorphic joint cybersecurity-channel coding scheme which was introduced for the first time in the first part of this article. Cybersecurity implementation is very important in modern wireless networks and communications since wireless data transactions is highly exposed to eavesdropping and malicious hacking. In addition, channel coding is highly important in wireless data transactions since channel coding is required to correct for data errors that are encountered due to the existence of unavoidable channel noise. Logic homeomorphism describes properties-preserving logic mapping that is intrinsically bijective, where it was introduced for the first time in the first part of this article that logic homeomorphism is utilized to achieve joint cybersecurity-channel coding implementation, where it was also shown that the implementation of the property of logic homeomorphism within cybersecurity via new symmetric key $k_s(k, N, n)$ can further enhance confidentiality and within channel coding can further enhance data integrity (thus better reliability) for the correction of many-error situations that are usually uncorrectable. Lattice grids possess the important property of high regularity which was proven to be very useful in low-power fault testing and localization, self-repair, size compactness and ease of manufacturability. Thus, the introduced nano-based regular lattice implementation of the newly introduced joint cybersecurity-channel coding method will be useful for improving system characteristics such as enhancing cybersecurity and error-correction capabilities, fault testing and localization, size compactness, speed improvement and the minimization of power con-

sumption. Applications of the new nano-based regular homeomorphic architectural design will include lower-power higher-performance and smaller-size designs of ASIC circuits and systems for higher data confidentiality and better data integrity (thus better reliability) within wireless data networking and transactions.

Keywords

Cybersecurity, Error-Control Coding, Lattice Grids, Logic Homeomorphism, Nano-Based Devices and Circuits

1. Introduction

Wireless data transactions usually encounter intense occurrence of noise that corrupts sent data messages, and thus noisy corrupted messages are usually received [1]-[4]. The source of corrupting noise is usually originated from the transmission medium, and therefore stream-based error correction and block-based (e.g., homeomorphic) error correction are highly important tasks in situations where channel noise occurs [1]-[4]. In order to solve the classical error detection and correction (EDAC) problem, several solutions have been implemented that include solving for error-control by using the methods of parity checking and utilizing various coding schemes that work optimally for specific kinds of statistical noise distributions such as convolution-based Viterbi and Turbo coders.

In modern cybersecurity, security goals are comprised of confidentiality, integrity and availability also called the C.I.A. triad. In general, the classification of security attacks with relation to security goals can be categorized as follows [5]-[7]: 1) the set of passive attacks {snooping, traffic analysis} are threats to confidentiality, 2) the set of active attacks {modification, masquerading, replaying, repudiation} are threats to integrity, and 3) the set of active attacks {denial of service} is a threat to availability. On the other hand, the relationship between security services and security mechanisms can be categorized as follows [5]-[7]: 1) the set of security mechanisms {encipherment, routing control} can be used to provide the security service of data confidentiality, 2) the set of security mechanisms {encipherment, digital signature, data integrity} can be used to provide the security service of data integrity for anti-change and anti-replay, 3) the set of security mechanisms {encipherment, digital signature, authentication exchanges} can be used to provide the security service of authentication for peer entity and data origin, 4) the set of security mechanisms {digital signature, data integrity, notarization} can be used to provide the security service of non-repudiation for proof of origin and proof of delivery, and 5) the set of security mechanisms {access control mechanism} can be used to provide the security service of access control. While security mechanisms are only a theoretical guide to implement security, security techniques are the actual implementation of security goals that include the general technique of cryptography which consists of encipherment (both symmetric-key and

asymmetric-key) and hashing, and the specific technique of steganography. For example, while cryptography aims to enciphering transacted data to protect message content by altering the data structure into ciphertext making it unreadable without a key and steganography aims to hiding the very existence (presence) of a secret one-to-one private message by embedding it inside another non-suspicious file, the other method of watermarking aims to establishing ownership and embedding a copyright (identity, mark) in a file within one-to-many public distribution.

The objectives regarding the C.I.A. triad can be jeopardized by security breaches, necessitating the use of general security methods like cryptography and specific techniques such as steganography to effectively implement these security goals. In cryptography, securing messages against attacks involves using three main mechanisms: 1) symmetric-key encryption, which employs a single key for both encrypting and decrypting, as seen in AES, DES and RC4 algorithms, 2) asymmetric-key encryption, utilizing a pair of keys—one public for encryption and one private for decryption—such as in RSA, Diffie-Hellman, and Elliptic Curve algorithms, and 3) hashing, where a fixed-size message digest is generated from a variable-length message, allowing both the original message and its digest to be sent to the recipient in applications requiring checkvalues, particularly in data integrity contexts.

Computing at the nanoscale is set to play a greater role in creating smaller and more energy-efficient computers in present and forthcoming technologies [8]-[10]. Several nanoscale computer circuits and systems must exhibit logical homeomorphism, particularly in nanoscale quantum computing systems [8] [9]. Consequently, homeomorphic computing will play a progressively larger role in the future development of standard, compact and universal circuits and systems. A (q, q) homeomorphic circuit is one with an equal number of inputs q and outputs q , establishing a one-to-one correspondence between the input and output vectors. Consequently, the input state vector can be distinctly reconstructed from the output state vector, making homeomorphic circuits free of information loss.

The first part of this article had demonstrated, for the first time, the effectiveness of logic homeomorphism in obtaining improvements for 1) confidentiality and 2) data integrity (and consequently reliability) through the newly proposed logically homeomorphic joint cybersecurity-channel coding design. Additional reasons for exploring the potential of employing circuits and systems through logic homeomorphism involve:

1) speed: because closed quantum unitary evolution is entirely homeomorphic (reversible), substantial improvements in computational speed will result when the quantum characteristics of superposition and entanglement in these nanoscale quantum mechanical systems are leveraged in the creation of computational circuits and systems.

2) power: internal calculations in homeomorphic systems are theoretically power-free, as was demonstrated [11] that *information is physical* with the heat dissipated for each non-homeomorphic bit operation expressed as $K \cdot T \ln(2)$, where K

being the Boltzmann constant and T the operating temperature, and a necessary (albeit insufficient) condition to avoid power dissipation in any physical system is that all circuits within the system need to be logically homeomorphic. Theoretically, fully homeomorphic digital systems will eradicate power consumption by meeting three criteria: a) logical homeomorphism: the input states vector can always be distinctly reconstructed from the output states vector, b) physical homeomorphism: the physical switch functions both backwards and forwards, and c) “ideal-like” switches that lack parasitic resistances.

3) size: present trends concerning nanoscale hardware applications are progressing toward atomic dimensions where quantum phenomena like entanglement and superposition are crucial, therefore the nascent nanoscale quantum computing paradigm must be logically homeomorphic (*i.e.*, reversible) due to the reality that closed quantum state evolution is entirely unitary.

Lattice grids [8] [12]-[14] extend the concepts from familiar regular circuits like spectral transform decision trees and decision diagrams, fat trees, generalized PLAs, Maitra cascades, and Akers arrays into a more organized framework that is intimately connected to the symmetry of functions and symmetric networks [8]. Lattice grids exhibit the crucial characteristic of regularity, which is beneficial for low-power fault testing and localization, self-repair, compact size and ease of manufacturing. In today’s intricate VLSI and ASIC system designs, standard interconnects often result in cost-effective implementations and elevated densities, where greater density signifies enhanced performance and reduced overhead for supporting components. Conventional circuit configurations that include the implementation of functions in three dimensions can be highly significant, as they demonstrate that synthesizing functions in three dimensions is optimal when all regular local interconnections are uniform in length and global interconnections are solely inputs on parallel oblique planes [8] [12]-[14]. Consequently, synthesis techniques utilizing two-dimensional and three-dimensional lattice circuits and systems have been explored and have been implemented in significant areas like nano-based controlled-switching architectures [15]-[17] and quantum computing [8].

The main contribution of this second part of the article is the introduction of a new nano-based architectural design of the novel homeomorphic joint cybersecurity-channel coding scheme that was introduced for the first time in the first part of the article, using regular lattice grids via nano-based carbon multiplexing devices and techniques. This involves the complete design of nano-based circuits for the implementations of the cybersecurity symmetric key-generated homeomorphic mapping Algorithm_n at the sender and receiver sides, homeomorphic encoder at the sender side, and the corresponding homeomorphic Viterbi decoder at the receiver side. **Figure 1** shows the details of the introduced system hierarchical implementations which are performed in this article.

The rest of this second part of the article is organized as follows: Important background on cybersecurity within modern wireless communication systems is presented in Section 2. Basic and important background on regular lattice grids is

presented in Section 3. Background on controlled-switching and many-valued computing using carbon nanotubes and carbon field emission is presented in Section 4. The nano-based circuit design for the homeomorphic joint Viterbi decoder-symmetric-key generated Algorithm_m at the receiver side as an important implementation example of the newly introduced homeomorphic joint cybersecurity-channel coding scheme is introduced in Section 5. Conclusions and future work are presented in Section 6.

Homeomorphic Joint Cybersecurity – Channel Coding Computation
Functional Lattice Grid Synthesis
Circuit and System Design
Carbon Controlled-Switching Devices
Solid-State, EM and Field-Emission Physics
Galois Field Algebra

Figure 1. Hierarchy layers of the introduced system implementation.

2. Cybersecurity in Modern Wireless Communication Systems

In general, modern wireless data transactions usually follow the stages shown in **Figure 2** for data transmission within which: 1) data is originated from a data source such as audio, image or sensors, 2) filtering produces cleaned data by removing noise to improve signal-to-noise ratio using appropriate filtering methods, 3) signal processing provides signal enhancements such as improving image quality, 4) source coding uses information theory to produce data compression to save bandwidth and achieve higher data rates, 5) data cybersecurity produces data encryption using cybersecurity techniques, 6) channel coding uses coding theory for error detection and correction to present redundancy into data in order to improve data integrity and thus data reliability, 7) signal coding converts data into the required form which is needed for signal transmission and medium requirements, 8) spread spectrum produces bandwidth increase of signal to be transmitted to enhance security, interference resistance, anti-jamming and reducing multipath fading, 9) multiplexing produces data streams from several sources that need to be transmitted over single medium and networking provides device networking over a network architecture, and 10) antenna provides interfacing into the utilized transmission medium. For data reception, wireless communication systems usually follow the reverse-order and inverse-function for the corresponding stages shown in **Figure 2**.

One of the biggest risks to computer security is unauthorized access to a computer system or network [5]-[7]. In order to avoid or minimize harm, intrusion detection systems have been created to provide early warning of an intrusion. Finding anomalous activity patterns or activity patterns that are known to be associated with intrusions is the goal of intrusion detection. Password management is an important example component of intrusion prevention which aims to keep unauthorized individuals from accessing other people's passwords. For example,

malicious software is intentionally inserted into a system for spying or to cause a harm. A virus, for example, is a program that infects other software by modifying it to include a copy of itself, allowing it to spread further. Another example is a worm, which is a self-replicating program that sends copies of itself to other computers across network connections; once it reaches a new system, the worm can activate to replicate and propagate again, while typically performing additional unwanted functions.

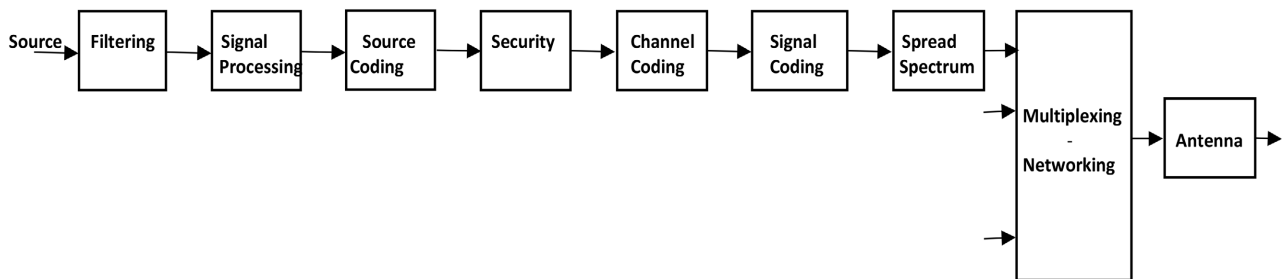


Figure 2. Stages of modern wireless data transmission: processing stages from data source to signal transmission where data is originated from a data source, filtering includes removing noise, signal processing provides signal enhancements, source coding provides data compression, security uses methods such as data encryption, channel coding includes error detection and correction, signal coding includes conversion of signal form, spread spectrum includes bandwidth spreading, multiplexing is used for transmission from multiple sources over a single medium and networking consists of device networking over a network architecture, and antenna is used for the interfacing into the utilized transmission medium.

In cybersecurity, a security attack refers to any act that undermines the information security of an organization [5]-[7], with these attacks categorized as passive attacks including unauthorized access to a message (file) and traffic analysis, active attacks which involve altering messages or files, and denial of service (DoS) attacks aimed at preventing legitimate users from accessing a service, with a distributed DoS originating from multiple coordinated sources.

A security mechanism refers to a procedure (or a device that includes such a procedure) which is intended to identify, avert or recover from a security breach, with examples of mechanisms including encryption algorithms, digital signatures, and authentication protocols [5]-[7].

A security service is a processing or communication service that enhances the security of an organization's data processing systems and information transfers. These services, which include authentication, access control, data confidentiality, data integrity, non-repudiation and availability, are designed to counter security attacks by utilizing one or more security mechanisms.

Message authentication is a mechanism used to verify the integrity of a communicated message [5]-[7]. It ensures that the received data remains unchanged—free from modifications, insertions, deletions, or replay attacks—and confirms the sender's identity. Symmetric encryption facilitates this authentication between parties sharing a secret key, while encrypting a message with a sender's private key serves as another form of authentication. Message authentication primarily relies on two cryptographic techniques: message authentication codes (MACs)

and secure hash functions. A MAC is an algorithm that utilizes a secret key; by processing a variable-length message and the key, it generates an authentication code. Recipients holding the same secret key can produce this code to verify the message's integrity. Conversely, a hash function maps a message of any length into a fixed-length hash value, known as a message digest. To be used for authentication, a secure hash function must be integrated with a secret key.

A digital signature is another form of authentication that allows a message creator to append a unique code, serving as a signature [5]-[7]. This signature is generated by hashing the message and encrypting it with the sender's private key, thereby ensuring both the message's source and its integrity. Mutual authentication protocols allow communicating parties to verify each other's identities and exchange session keys. Conversely, one-way authentication provides the recipient with assurance regarding the sender's identity. For example, the Digital Signature Standard (DSS) is a NIST-established protocol that utilizes the Secure Hash Algorithm (SHA).

On the other hand, and in terms of network security, the Secure Sockets Layer (SSL) provides security services for applications utilizing TCP [5]-[7]. Its internet standard version is known as Transport Layer Security (TLS). These protocols ensure confidentiality through symmetric encryption and maintain message integrity using a message authentication code. Furthermore, SSL/TLS incorporates mechanisms that allow two TCP users to negotiate the specific security services and protocols they will employ.

Symmetric encryption, also referred to as conventional encryption, is a cryptographic method that utilizes the same key for both encryption and decryption. In this process, plaintext is converted into ciphertext through an encryption algorithm and a secret key [5]-[7]. To retrieve the original information, the same key is applied alongside a decryption algorithm to revert the ciphertext back into plaintext. Attacks on these systems generally fall into two categories: a) cryptanalysis which exploits the technical properties of the encryption, and b) brute-force which systematically tests all potential keys. Traditional pre-computer symmetric ciphers rely on substitution and transposition methods. Substitution maps plaintext characters into ciphertext elements, while transposition systematically rearranges the positions of those characters. Rotor machines, for instance, are complex pre-computer hardware devices that employ substitution techniques. In contrast, the more specific technique of steganography involves concealing a secret message within a larger one to ensure that its presence remains undetectable to others while for example watermarking targets establishing ownership by embedding a copyright mark in a file.

A stream cipher is a symmetric encryption algorithm that processes plaintext input byte-by-byte to generate ciphertext output, with RC4 being the most prominent example. In contrast, a block cipher encrypts entire blocks of plaintext to produce ciphertext blocks of the same size [5]-[7]. Many block ciphers utilize a Feistel structure, which involves multiple identical rounds of processing. During

each round, one half of the data undergoes substitution, followed by a permutation that swaps the two halves. The original key is expanded to provide a unique sub-key for every round. Until recently, the Data Encryption Standard (DES), which employs a 64-bit block size and a 56-bit key, was the most widely used encryption algorithm and served as a classic implementation of the Feistel structure. Additionally, differential and linear cryptanalysis are two significant methods used for breaking these ciphers, where DES has proven highly resistant to these two types of attacks. AES, a block cipher designed to succeed DES in commercial applications, utilizes a 128-bit block size and supports key lengths of 128, 192, or 256 bits. Unlike DES, AES does not employ a Feistel structure; instead, each full round incorporates four distinct functions: byte substitution, permutation, finite field arithmetic, and XOR operations with a key. Multiple encryption involves applying an encryption algorithm repeatedly, where the ciphertext from one stage serves as the input for the next. This process can be extended across any number of stages. Triple DES, for example, utilizes three stages of the DES algorithm, employing either two or three distinct keys. On the other hand, a mode of operation is a method for improving the impact of a cryptographic algorithm or modifying the algorithm for a particular use, like applying a block cipher to a stream of data or a series of data blocks. For use with symmetric block ciphers like DES and AES, five standard operating modes have been established: electronic codebook mode, cipher block chaining mode, cipher feedback mode, output feedback mode, and counter mode.

Asymmetric encryption is a type of cryptosystem where different keys—a public key and a private key—are used for encryption and decryption [5]-[7]. Another name for it is public-key encryption. Asymmetric encryption uses an encryption algorithm and one of two keys to convert plaintext into ciphertext. The plaintext is extracted from the ciphertext using a decryption algorithm and the paired key. Confidentiality, authentication, or both can be achieved with asymmetric encryption. For example, RSA is the most popular public-key cryptosystem. The challenge of determining a composite number's prime factors is the basis for the difficulty of RSA attacks. Only when the public key's authenticity is guaranteed can public-key encryption schemes be considered secure. The required security is provided by a public-key certificate scheme. Another example is the Diffie-Hellman key exchange which is a basic public-key algorithm. Using a public-key scheme based on discrete logarithms, this protocol allows two users to create a secret key. Only when the two participants' identities can be verified is the protocol considered secure. Key exchange, encryption, and digital signatures are just a few of the elliptic curve cryptography (ECC) schemes that can be created using elliptic curve arithmetic [5]-[7]. An elliptic curve equation defined over a finite field is used in elliptic curve arithmetic for ECC purposes where the equation's variables and coefficients are components of a finite field.

As an application example, PGP is a free and open-source email security program that uses digital signatures for authentication, symmetric block encryption

for confidentiality, ZIP algorithm for compression, radix-64 encoding scheme for email compatibility, and segmentation and reassembly to handle lengthy emails [5]-[7]. PGP includes tools for managing public-key certificates and creating a public-key trust model. The same features as PGP are included in S/MIME which is another internet-standard method of email security.

3. Lattice Grids

As logic implementation in present and upcoming technologies becomes increasingly miniaturized, attention will progressively shift towards interconnected issues such as regularity, compactness, predictable timing, manufacturability, enhanced testability, quick fault localization, and self-repair capabilities. In today's top technologies with device quantities in the billions, over 80% of circuit regions are typically filled with local and global interconnects, leading to interconnect delays accounting for 50% or greater of the total circuit delay. In upcoming technologies, interconnects will occupy an even larger percentage of area and delay, leading to growing interest in cellular and regular circuits, particularly for nanotechnologies. **Figure 3** depicts the trends in signal delays for global interconnects compared to local interconnects [18].

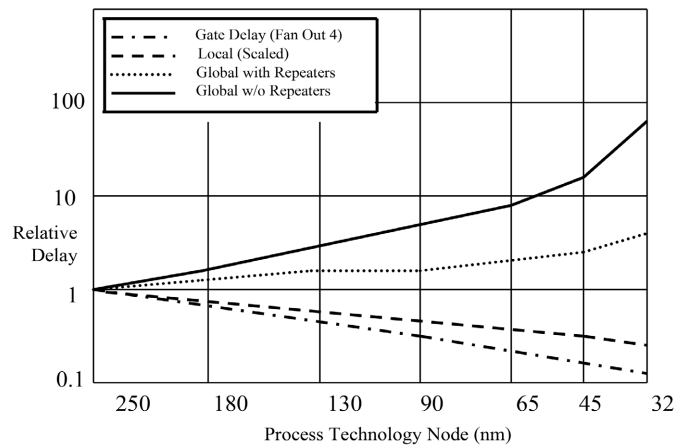


Figure 3. Delays for local and global interconnects versus the utilized feature size.

Lattice grids [8] [12]-[14] expand upon concepts such as spectral transform decision trees and diagrams, fat trees, generalized PLAs, Maitra cascades and Akers arrays, forming a more organized framework closely connected to the idea of *logic symmetry* [8]. Additionally, it has been shown that implementing logic circuits in three-dimensional space could be crucial for future technologies since it demonstrates that the optimal method to logically synthesize combinational functions is within solid three-dimensional space, where all local connections are of uniform length and global connections are solely inputs on parallel inclined planes [8]. For instance, as greater power consumption arises in contemporary IC circuit design with the increase of global interconnects over local ones, lattice grids provide an effective solution to the

issue of rising IC power consumption because they utilize solely local interconnects [8].

3.1. Logic Symmetry

In logic synthesis, it is widely recognized that particular categories of logic functions display distinct forms of symmetries [8]. These symmetries are present in various forms, including symmetries among distinct functions when negated, symmetries in a function regarding the negation of its variables, and symmetries in a function concerning the permutation of its variables. A technique to define a symmetry that could be present in a logic function is executed by employing the concept of symmetry indices. A symmetry index S^i specifies a K-map cell that counts the value of “1” in the specified minterm for number of times equal to i .

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

Definition 2. Elementary symmetric functions of n variables are:

$$\begin{aligned} S^0 &= \bar{x}_1 \bar{x}_2 \cdots \bar{x}_n, \\ S^1 &= x_1 \bar{x}_2 \cdots \bar{x}_n + \bar{x}_1 x_2 \bar{x}_3 \cdots \bar{x}_n + \cdots + \bar{x}_1 \bar{x}_2 \cdots x_{n-1} x_n, \\ &\dots \\ S^N &= x_1 x_2 \cdots x_n. \end{aligned}$$

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

It has been demonstrated that a non-symmetric function can be made symmetric by duplicating variables [8]. This variable duplication method alters the values of cells in the related K-map (that cause the function to be nonsymmetric) into don't cares and thus rendering the function symmetric [8]. For example, it can be observed that although differing values may arise for a symmetry index in minterms in a K-map, resulting in a non-symmetric function, consistent values are generated for that same non-symmetric function through the technique of variable repetition.

As mentioned earlier, different uses of symmetry indices for the creation of logic functions have been demonstrated before [8]. This encompasses symmetric networks, Akers arrays, and lattice grids, along with various other implementations. The idea of lattice grids for switching functions typically includes three elements: 1) the expansion of a function which is linked to the root in the lattice producing multiple successor nodes of the expanded node, 2) the merging of several nodes into one node which is the inverse of the expansion operation, and 3) a regular geometry to which the nodes are assigned. **Figure 4(a)** illustrates a four-variable (*i.e.*, four-level) two-dimensional lattice grid featuring inner nodes

as two-to-one multiplexers, while **Figure 4(b)** and **Figure 4(c)** depict the connection between **Figure 4(a)** and symmetry indices [8]. **Figure 4** illustrates that conventional lattice grids demonstrate a strong connection with symmetry indices, which denote the collections of all potential paths from leaves to the root of the respective lattice grid. It is also observed that every internal node in **Figure 4(a)** represents a Shannon node that is effectively realized as a two-to-one multiplexer directing its output in two ways, where different types of expansion nodes can also be generated as well that yield alternative representation forms of lattice grids [8].

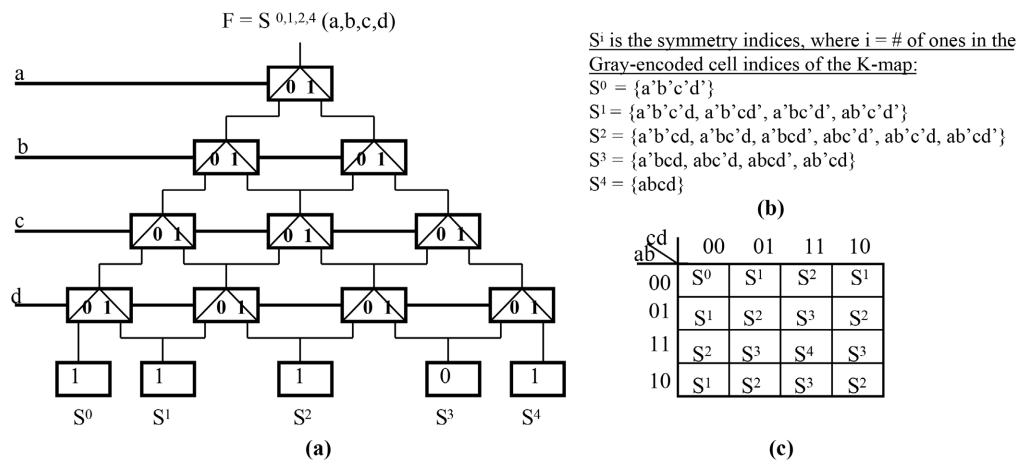


Figure 4. The connection between a two-dimensional lattice grid and symmetry indices: (a) a two-dimensional lattice grid for a four-variable function F , (b) groups of binary symmetry indices, and (c) the related K-map representation of the binary symmetry indices.

It has been demonstrated, as noted earlier, that any non-symmetric function can be symmetrized by duplicating some of its variables. Consequently, because each variable corresponds to one level in the lattice grid, duplicating variables results in repeated levels within the lattice grid. Typically, three primary factors determine the size of a lattice grid that implements non-symmetric functions: 1) types of expansions utilized at the internal nodes, 2) arrangement of variables upon which functions are expanded at every level of the lattice, and 3) selection of repeated variables [8]. As a result, several optimization techniques have been discussed for the best selection of these three factors stated earlier to reduce the size of the lattice grid for implementing the associated logic functions.

It is shown that to maintain the standard realization of expansions over n radix, it is adequate to connect n nodes in n -dimensional space to achieve the appropriate lattice grids [8]. For example, in the binary case, connecting two nodes is enough to create two-dimensional lattice grids, while in the ternary case, linking three nodes is adequate to generate the associated three-dimensional lattice grids. Similar to the binary scenario, completely symmetric ternary functions don't require duplicate variables for their implementation in three-dimensional lattice grids. **Figure 5** illustrates the connection between three-dimensional lattice grids and ternary symmetry

indices, where ternary symmetry indices consist of all potential paths from the leaves to the root of a three-dimensional lattice grid [8]. Observe that every internal node in **Figure 5** represents a ternary Shannon node functioning as a three-to-one multiplexer with its output directed in three different ways.

As demonstrated earlier, implementing non-symmetric functions with standard lattice grids requires the technique of variable repetition [8]. Often, variables must be repeated so frequently that it leads to a significantly large lattice grid, which may not conform to the designated area or the defined volume in the context of three-dimensional lattice grids [14]. Conversely, one can adjust the relevant interconnections among the internal nodes of the lattice grid through optimization techniques to ensure that the grid conforms to the designated spatial arrangement. However, this re-routing procedure will result in the interconnects among lattice nodes having varying lengths, thereby depriving the lattice grid of one of its key attributes of regularity where all interconnects should be of uniform length.

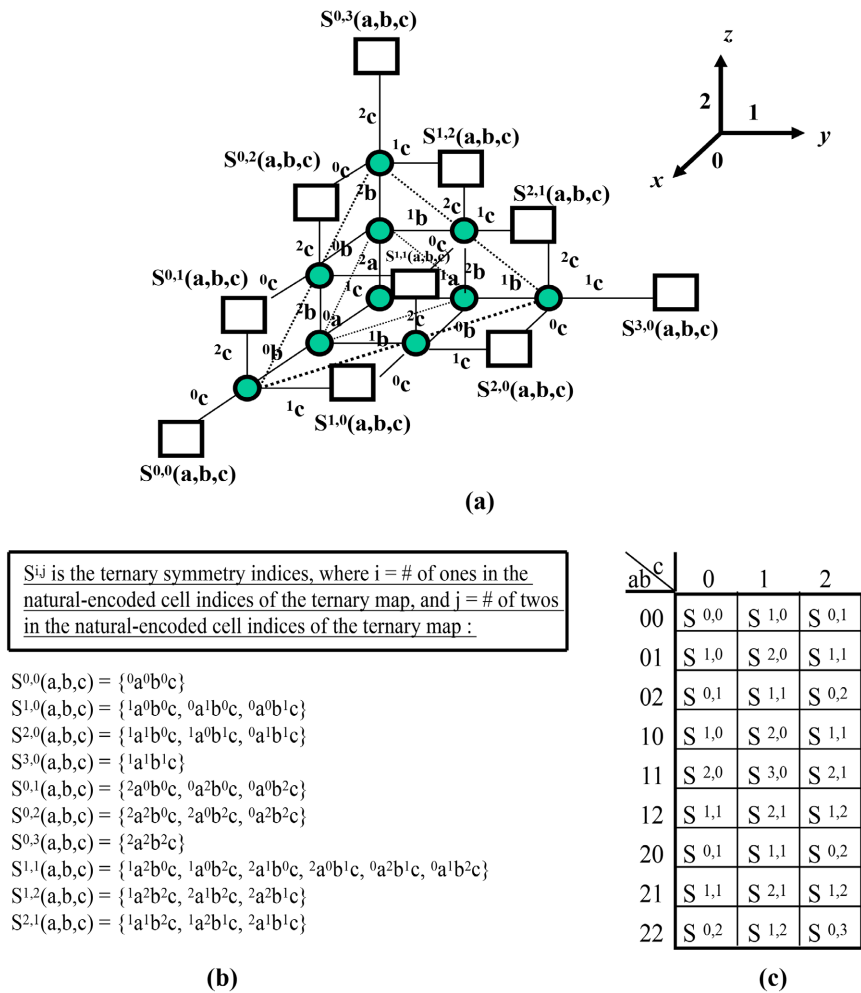


Figure 5. Three-dimensional lattice grid for three-variable ternary functions: (a) lattice grid, (b) ternary symmetry indices, and (c) ternary natural-coded representation for the ternary symmetry indices.

3.2. Fundamental Two-Dimensional Lattice Grids

The binary structure of families of canonical forms and their associated decision trees and diagrams relies on three essential functional expansions of Shannon, positive Davio and negative Davio expansions [8]. The basic Shannon expansion is presented in Equation (1):

$$f(x_1, x_2, \dots, x_n) = x_1' f_0(x_1, x_2, \dots, x_n) \oplus x_1 f_1(x_1, x_2, \dots, x_n) \quad (1)$$

where $f_0(x_1, x_2, \dots, x_n) = f(0, x_2, \dots, x_n) = f_0$ is the negative cofactor of variable x_1 and $f_1(x_1, x_2, \dots, x_n) = f(1, x_2, \dots, x_n) = f_1$ is the positive cofactor of variable x_1 . All operations in Equation (1) are performed using Boolean algebra where $\oplus$ is the Boolean XOR.

Due to the fact that Galois field (GF) has demonstrated multiple advantages in various applications like testing, communications and signal processing, the three-dimensional lattice grids will be carried out on the relevant GF. For instance, addition and multiplication operations in radix three GF are detailed in **Table 1(a)** and **Table 1(b)**, respectively.

Table 1. Third radix Galois field: (a) addition and (b) multiplication.

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

A literal is a function of a single variable. The important 1-Reduced Post literal (1-RPL) is defined as:

$${}_i x = 1 \text{ iff } x = i \text{ else } {}_i x = 0 \quad (2)$$

For example, ${}_0 x$, ${}_1 x$, and ${}_2 x$ are the zero, first and second polarities of the 1-RPL, respectively. Also, the ternary shifts of variable x are defined as x with no shift, x' with one shift and x'' with two shifts (*i.e.*, $x = x + 0$, $x' = x + 1$, and $x'' = x + 2$, respectively) and $x \in \{0, 1, 2\}$, where ternary 1-RPL will be used to construct the corresponding three-dimensional lattice grids by controlling the propagation of sub-functions in three-dimensions [8].

As mentioned earlier, the idea of regular lattice grids for switching functions encompasses three elements: 1) expanding a function that relates to the root (ini-

tial node) in the lattice resulting in multiple successor nodes from the expanded node, 2) joining several nodes at a level of a decision tree into one node, and 3) regular geometry to which the nodes are assigned. For instance, although the implementation of Boolean non-symmetric functions in Akers arrays necessitates an exponential increase in variable repetition in the worst-case scenario, the implementation of these functions in regular lattice grids involves a linear increase in variable repetition [8]. Thus, for most practical benchmarks, there is no need to repeat the variables of non-symmetric functions excessively to implement such functions in regular lattice grids.

Definition 3. The function produced by connecting two nodes (sub-functions) in a lattice grid is referred to as the joined function. The function produced in nodes apart from the joining nodes, to maintain the functionality within the lattice grid, is referred to as the correction function.

Figure 6 demonstrates, for instance, the structure of a four-neighbor regular lattice grid and the joining operations applied to the nodes, where each node possesses two inputs and two outputs (*i.e.*, four neighbors). The design of the lattice grid shown in **Figure 6** employs one potential method of top-to-bottom expansion and left-to-right joining.

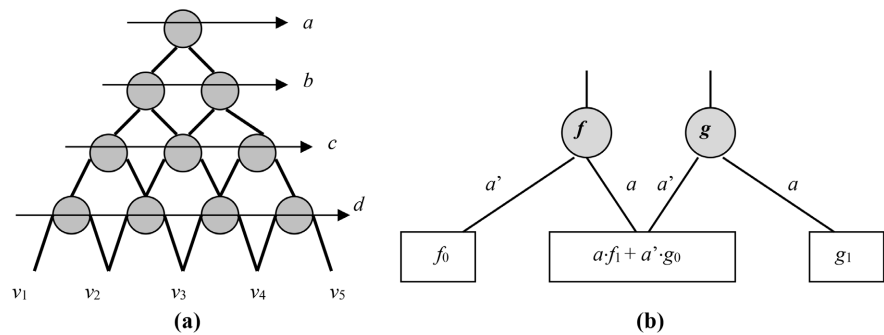


Figure 6. Two-dimensional four-neighbor lattice grids: (a) general lattice grid where $v_i \in \{0, 1\}$ and (b) Shannon lattice grid where internal nodes are 2-to-1 multiplexers.

Observe that the lattice grids depicted in **Figure 6** maintain the functionality of their respective sub-functions f and g . For example, as shown in **Figure 6(b)**, the complemented variable a' cancels variable a when cofactors are propagated from lower levels to upper levels or the other way around, without requiring correction functions to maintain the output functionality of the respective lattice grid. This straightforward observation may not directly be observable in other types of lattice grids because correction functions can be needed to negate the impact of the newly added nodes to maintain the functionality of the new lattice grids. As stated previously, it has been demonstrated that any non-symmetric function can become symmetric through variable repetition, and a completely symmetric function can be derived from any arbitrary non-symmetric function by repeating its variables. As a result, standard lattice grids and function symmetry are closely interconnected [8]. Example 1 illustrates this extremely close connection. It can be noted

that to depict the non-symmetric function from Example 1 in the two-dimensional regular lattice grid, variable a appears two times, with nodes in **Figure 7** being Shannon nodes that function as two-input one-output multiplexers directing their output in two directions, and variables $\{a, b\}$ serve as control signals.

Example 1. For the binary non-symmetric implication function $F = a' + b$, **Figure 7** illustrates the relationship between the K-map with non-conflicting symmetry indices S^i and the two-dimensional regular lattice grid with non-conflicting leaves, which is achieved by repeating variable a twice in the two-dimensional regular lattice grid. All internal nodes in **Figure 7** are two-to-one multiplexers where multiplying each leaf value from left to right with every possible bottom-up path (from leaves to root) and summing them using Boolean algebra produces the desired function F at the root.

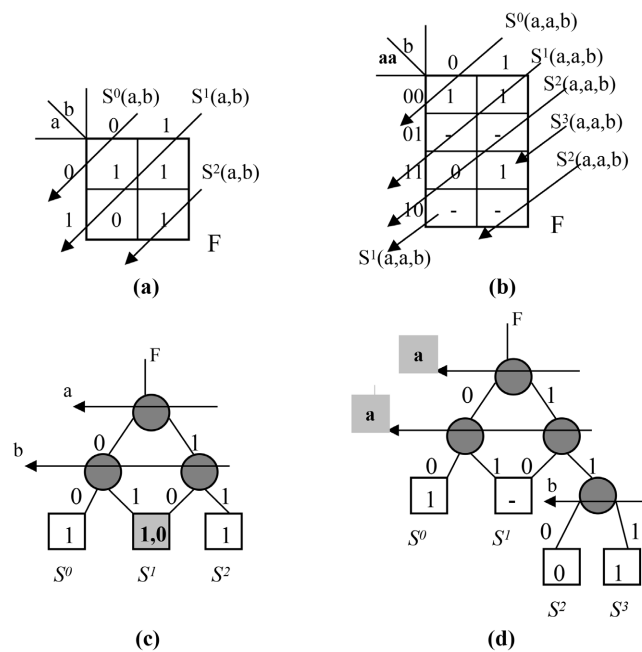


Figure 7. Employing symmetry indices: (a) non-symmetric implication Boolean function, (b) symmetrization through the repetition of variable a , (c) two-dimensional regular lattice aligned with **Figure 7(a)** featuring a conflicting leaf in a dark box, and (d) two-dimensional regular lattice corresponding to **Figure 7(b)** with non-conflicting leaves.

Proving that repetition of variables will conclude during the symmetrization of non-symmetric functions is essential. A straightforward proof is as follows: the count of variables for completely symmetric functions matches the count of levels in the lattice grid since variable repetition is unnecessary. It is established that the repetition of variables symmetrizes every non-symmetric function, which consequently leads to a specific number of levels in the associated lattice grid, resulting in a certain number of total variables (both repeated and non-repeated) that conclude the symmetrization process.

The findings from this Subsection will be applied in the subsequent Subsection 3.3 to ternary logic, transitioning from two-dimensional planar space to three-

dimensional solid space.

3.3. Generalization to Three-Dimensional Lattice Grids

Many-valued logic holds a significant position between two-valued logic (like Boolean logic) and infinite-valued logic (like fuzzy logic) with numerous applications in contemporary communications and digital computing [8]. For instance, the idea of two-dimensional regular lattice grids that was mentioned in Subsection 3.2 can be expanded to encompass three-dimensional lattice grids [8]. As the most intuitive approach to conceptualize a binary lattice grid is through a four-neighbor two-dimensional lattice like that illustrated in **Figure 6(a)**, this concept can be expanded to three-dimensional space for ternary lattice grids, where these signify six-neighbor three-dimensional lattices [8].

While regular lattice grids can be constructed in three-dimensional space for the third Galois radix while preserving their complete regularity, they cannot be realized for radices greater than three; higher-dimensional lattice grids can be implemented in three-dimensional space but at the cost of forfeiting complete regularity. The ternary circuit realization yields a regular circuit in three dimensions, characterized by uniform interconnects of equal length; however, constructing higher-dimensional lattice grids within lower-dimensional space can sacrifice regularity, resulting in lattice grids that lack full regularity because of the varying lengths of interconnects connecting nodes.

As a geometric idea, and as mentioned before, lattice grids can be formed for two, three, four or any greater Galois bases. Nonetheless, since our physical environment is three-dimensional, lattice grids can only be constructed within a solid space exclusively for a) radix two in two dimensions or b) radix three in three dimensions. It is noteworthy that the regularity which is seen in lattice grids for a) binary and ternary functions will not be apparent for quaternary or higher radix functions. Consequently, ternary lattice grids occupy a distinctive position as extremely organized frameworks that optimize the use of three-dimensional space [8] [12]-[14]. As illustrated in Example 1, each dimension in a three-dimensional lattice grid relates to a value of the associated control variable where a value of zero for the control variable spreads along the x -axis, a value of one spreads along the y -axis, and a value of two spreads along the z -axis.

Example 2. This example illustrates the implementation of a ternary 3-digit full adder utilizing three-dimensional regular lattice grids. In this example, three-valued addition is executed using the modsum operator from **Table 1(a)**, which demonstrated the modsum addition applied to ternary inputs. The ternary maps depicted in **Figure 8** illustrate the ternary Sum F_1 and the ternary output Carry F_2 functions where: ternary Sum $F_1 = {}^0a^0b^1c + {}^0a^1b^0c + 2 \cdot {}^0a^1b^1c + 2 \cdot {}^0a^2b^0c + {}^1a^0b^0c + 2 \cdot {}^1a^0b^1c + 2 \cdot {}^1a^1b^0c + {}^1a^2b^1c + 2 \cdot {}^2a^0b^0c + {}^2a^1b^1c + 2 \cdot {}^2a^2b^0c + 2 \cdot {}^2a^2b^1c$, and ternary output Carry $F_2 = {}^0a^2b^1c + {}^1a^1b^1c + {}^1a^2b^0c + {}^1a^2b^1c + {}^2a^0b^1c + {}^2a^1b^0c + {}^2a^1b^1c + {}^2a^2b^0c + {}^2a^2b^1c$. The displayed representation is executed by employing 1-RPLs for the variables $\{a, b, c\}$ and utilizing the relevant ternary Galois field addition and multiplication

operations from **Table 1**. The lattice grids depicted in **Figure 9** and **Figure 10** illustrate the three-dimensional regular lattice grid representations of the ternary maps in **Figure 8**, with the symbol - indicating ternary don't care $\in \{0, 1, 2\}$.

As seen in Example 2, **Figure 9** and **Figure 10** illustrate a completely regular lattice grid in three dimensions. Every dimension relates to a value of the respective control variable, with value zero extending along the x -axis, value one extending along the y -axis and value two extending along the z -axis. Because the ternary function in Example 2 is symmetric, it was unnecessary to repeat any variables in the related regular lattice grid. In three-dimensional space, every control variable extends across a plane to manage the respective nodes, with these parallel planes illustrated by the appropriate dotted triangles as demonstrated in **Figure 9** and **Figure 10**. This differs from the binary scenario, where each control variable extends along a horizontal line to regulate the associated nodes, as illustrated in **Figure 6(a)**. Also, each node in **Figure 9** and **Figure 10** represents a three-input one-output multiplexer whose output goes in three directions.

Every internal node in **Figure 9** and **Figure 10** functions as a three-to-one multiplexer where multiplying each leaf value—moving counterclockwise—by all potential out-to-in paths (from leaves to root) and summing them using GF operations from **Table 1** yields the maps F_1 and F_2 found in **Figure 8(a)** and **Figure 8(b)**, respectively. Additionally, it is observed that internal nodes in **Figure 9** and **Figure 10** are positioned at the corners of three-dimensional cubes, unlike the binary case where nodes are situated at the corners of two-dimensional squares. As seen, the Sum and output Carry functions display symmetry, as there are no discrepancies in leaf values, eliminating the necessity to duplicate variables for these ternary functions to be implemented in the relevant three-dimensional regular lattice grids. Conversely, at least one leaf contains conflict values within non-symmetric ternary functions, hence the duplication of variables to symmetrize the related non-symmetric functions for implementing these functions in three-dimensional lattice grids.

a b \ c		c	
		0	1
0	0	0	1
0	1	1	2
0	2	2	0
1	0	1	2
1	1	2	0
1	2	0	1
2	0	2	0
2	1	0	1
2	2	1	2

(a)

a b \ c		c	
		0	1
0	0	0	0
0	1	0	0
0	2	0	1
1	0	0	0
1	1	0	1
1	2	1	1
2	0	0	1
2	1	1	1
2	2	1	1

(b)

Figure 8. Ternary maps for (a) ternary Sum F_1 , and (b) ternary output Carry F_2 .

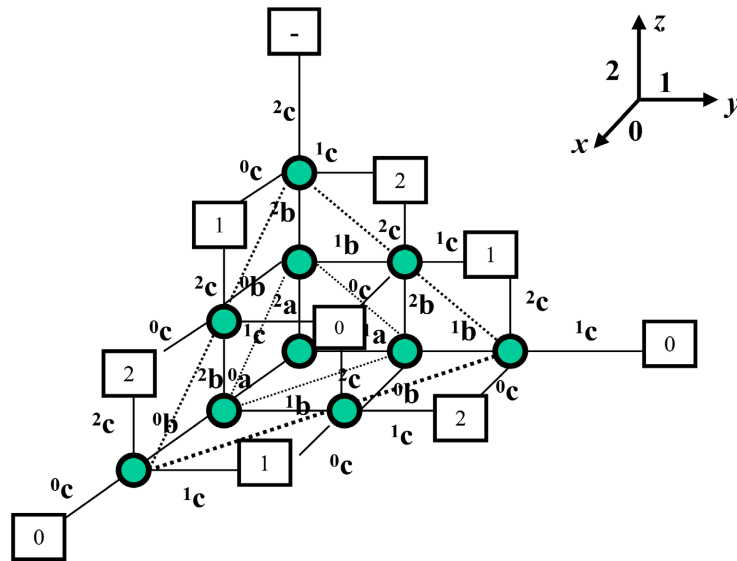


Figure 9. The Sum function F_1 of the ternary 3-digit full adder.

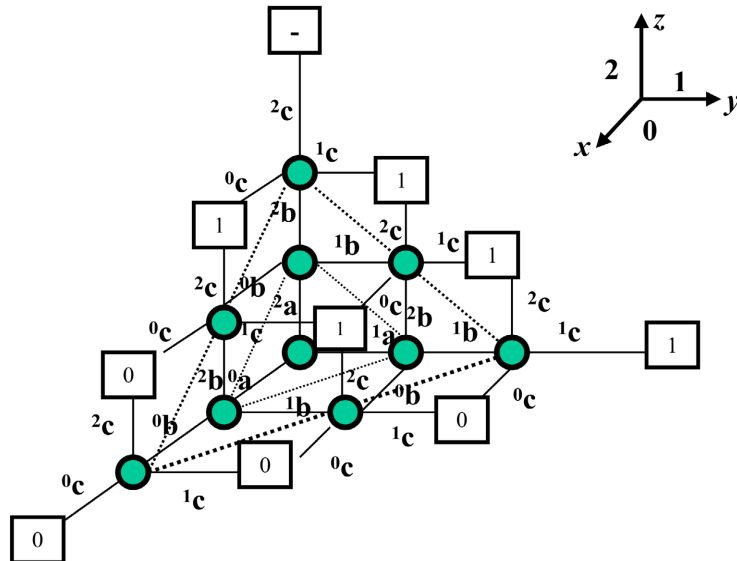


Figure 10. The output Carry function F_2 of the ternary 3-digit full adder.

Three-dimensional regular lattice grids can also be used to synthesize Boolean functions in three-dimensional space, where the synthesis of binary functions is performed in three-dimensional space by the application of function-preserving two-valued to three-valued mapping. For example, this can be done as one can observe that both independent and dependent variables in the two-valued logic can have only two values $\{0, 1\}$ and thus $f(x_1, \dots, x_n) = {}^0x_1 \cdot f_0(x_1, \dots, x_n) \oplus {}^1x_1 \cdot f_1(x_1, \dots, x_n)$ where ${}^i x$ is the two-valued 1-RPL for i and $x \in \{0, 1\}$, and both independent and dependent variables in the three-valued logic can have three values $\{0, 1, 2\}$ and thus $f(x_1, \dots, x_n) = {}^0x_1 \cdot f_0(x_1, \dots, x_n) + {}^1x_1 \cdot f_1(x_1, \dots, x_n) + {}^2x_1 \cdot f_2(x_1, \dots, x_n)$ where ${}^i x$ is the three-valued 1-RPL for both i and $x \in \{0, 1, 2\}$, and therefore any two-valued function can be represented in three-valued logic

by ignoring the second polarity and using only zero and first polarities. Thus, the two-valued function can be represented in three-dimensional lattice grid by ignoring the direction of the second polarity and using only the other two directions of zero and first polarities.

The realization of two-valued functions in three-dimensional lattice grid can be highly needed as 1) several important directions within near future technologies will perform function synthesis and the corresponding circuit fabrication in three-dimensional regular layout in order to minimize two-dimensional planar area space by fabricating the required IC circuits into the corresponding three-dimensional solid volume space, and 2) the ability to utilize the same three-dimensional lattice grid for the synthesis of three-valued functions besides the two-valued functions and thus achieving multi-functional operations using the same three-dimensional lattice grids.

Lattice grids that were presented in this Section will be later utilized in Section 5 to synthesize the corresponding digital functions for the important example of homeomorphic joint Viterbi decoder-symmetric-key cybersecurity where two-dimensional regular lattice grids will be used for the synthesis of binary functions in the same way that was shown in Subsection 3.2.

4. CNT-Based and CFE-Based Controlled-Switching and Computing

Nanotechnology is a multidisciplinary area of study that intersects various domains of engineering, physics, chemistry and biology, which examines and creates systems at the nanoscale including nanoparticles, nanowires, nanosheets, nanotips and carbon nanotubes (CNTs) [10] [19]-[22]. Examples consist of nanocapsules that house charged particles for improved medical delivery, fundamental logic gates like inverters and multiplexers employing CNTs as a channel, nanoelectronics found in nano integrated circuits, CNT field-emission which is applied in scanning electron microscopy, circuits and actuators based on CNT multiplexers, and carbon field-emission (CFE) using carbon fiber nano-apex tips.

This Section provides essential information on CNTs and CFEs, along with the relevant many-valued computations facilitated by the presented CNT-based and CFE-based multiplexing devices. This background information will be applied in Section 5 to develop nano-based regular lattice grids for the new homeomorphic joint cybersecurity-channel coding design.

4.1. CNT Properties

Recently, CNTs have garnered significant interest due to their distinct shapes, small sizes and unique properties, as well as their possible applications in various existing and developing technologies [10] [19]-[22]. The CNT is composed of graphite which can form at the nanoscale in three distinct varieties:

1) Carbon nanoball, also known as buckyball, which is a molecule made up of 60 carbon atoms (C₆₀) that are arranged in a shape resembling a soccer ball.

2) Carbon nanotube, which is a cylindrical nanostructure formed from a narrow strip of tiny graphite sheets, primarily existing in three forms: a) multi-walled CNT (MWCNT) where each CNT includes multiple hollow carbon cylinders nested together, b) single-walled CNT (SWCNT) consisting of a single layer of carbon tube, and c) double-walled CNT (DWCNT) which features two concentric layers and combines the attributes of both MWCNT and SWCNT exhibiting greater stability than SWCNT.

3) Carbon nanocoil (CNC) which is a three-dimensional helical spring-like form characterized by high elasticity, surface area and conductivity.

CNT technology is among the advanced emerging technologies within nanotechnology that has demonstrated high efficiency and diverse applications. Recent instances of these applications involve a) CFE-based CNT TVs that require significantly less power, thinner and offer superior resolution, b) CNT-based nanocircuits like CNT Field Effect Transistors (FETs) that possess great potential for reduced power consumption and increased speed compared to existing silicon-based FETs, and c) carbon nanocoils that can function as inductors in nanofilters and as mechanical nanosprings. In addition, CNTs have also been demonstrated for potentially exciting applications such as in CNT probes, new composite materials, CNT data storage devices capable of storing 10^{15} bytes/cm², drug delivery systems, nano lithography and CNT gears. The CNT characteristics of current carrying capacity, heat resistance, energy usage, and electron scattering make CNTs suitable for future applications such as in a) efficient power transmission systems, b) quantum computing for their ability to maintain the electron spin, c) nanowires which would lead to a reduction in the overall area taken up by interconnects during the fabrication of integrated circuits, d) constructing circuits and systems that must endure stress without sustaining structural damage, and e) various applications that need a broad spectrum of energy band gaps spanning from conductor to semiconductor states.

The CNT development, analyzed through Transmission Electron Microscopy (TEM), Atomic Force Microscopy (AFM) and Scanning Electron Microscopy (SEM), necessitates procedures with appropriate conditions and materials. At present, there are several techniques for producing different types of CNTs including 1) a significant spark formed between two graphite rods spaced a few millimeters apart, 2) chemical vapor deposition (CVD), and 3) a laser pulse directed at a graphite target.

4.2. CNT-Based Solid-State Multiplexer

Figure 11 illustrates a CNT-based solid-state multiplexer [10] in which CNT serves as the channel in a Field Effect Transistor (FET). In this context, we apply CNT as a FET channel in the device comprising of two CNT n-FETs and two CNT p-FETs with distinct interconnect geometries that serve for inputs *A* and *B*, controls *C* and *C'*, and output *F*. **Figure 11** illustrates silicon (Si) functioning as a back-gate in the device, silicon dioxide (SiO₂) acting as an insulator, and gold serv-

ing as an electrode (conductor). Four metallic catalyst islands, with each pair situated at the opposing ends of the gold electrodes, can be utilized to cultivate CNTs between these pairs of electrodes instead of merely positioning CNTs in contact with the gold electrodes. The PMMA acts as a shield that safeguards anything underneath from contact with oxygen (O_2), which is required to change an n-CNT-FET into a p-CNTFET.

The operation of the device depicted in **Figure 11** can be described in the following manner: If $C = "1"$, the upper transmission gate (t-gate) is engaged allowing A to pass to F while the lower t-gate is inactive preventing B from passing. Conversely, if $C = "0"$, the lower t-gate activates permitting B to pass to F while the upper t-gate is inactive thus blocking A . Hence, the CNT-based solid-state multiplexer functions precisely as a 2-to-1 multiplexer.

Intra-molecular CNTFET technology can replace each sub-device in **Figure 11**, where a single CNT bundle is positioned atop three gold electrodes to create n-type and p-type CNTFETs on the same substrate, utilizing the same approach shown for the inter-molecular CNTFET. The creation of every sub-device in **Figure 11** occurs as follows: At the outset, the two CNTFETs are of p-type. Following vacuum annealing, both CNTFETs transform into n-type. Both CNTFETs are subjected to oxygen for three minutes, resulting in the unprotected n-CNTFET reverting to its original p-type, while the protected CNTFET stays n-type. An alternative approach to create p-type and n-type CNTFETs involves doping a CNT channel with potassium to yield an n-type while a non-doped CNT results in a p-type. Doping produces n-type CNTFET by shifting the Fermi energy level to the conduction band that results in an increase of electron concentration in that band which increases the corresponding conductance of the FET for a given positive gate voltage.

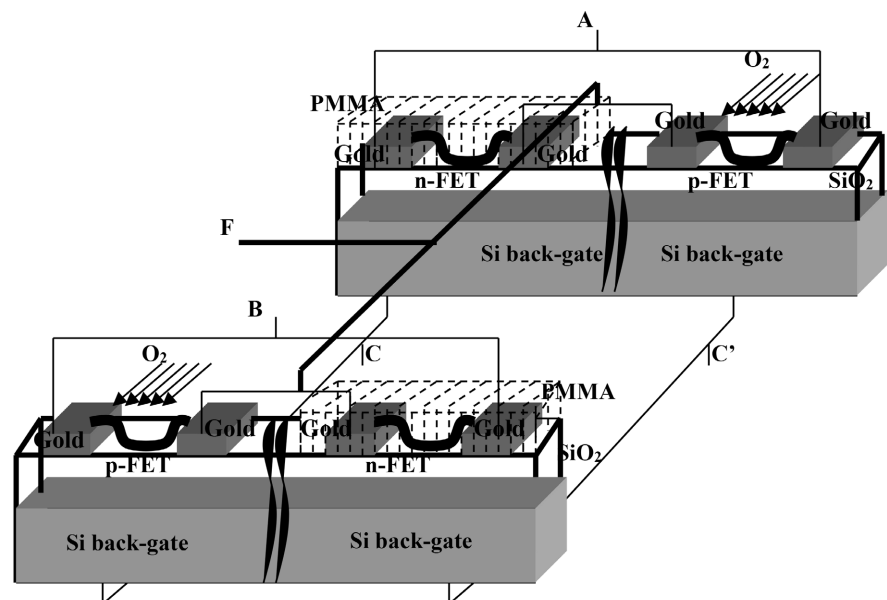


Figure 11. CNT-based solid-state multiplexer where CNT is used as FET channel.

4.3. CNT-Based Magnetic Multiplexer

Figure 12 depicts the architecture of the CNT-based magnetic multiplexer [10]. The device depicted in **Figure 12** operates with the electrical current (#7) serving as input B which can have two states representing logic “0” and “1”, correspondingly. Additionally, the electrical current (#8) utilized as input A can represent two states that denote logic “0” and “1”, correspondingly. The two electrical currents (#7) and (#8) are moving in the directions shown in **Figure 12** in CNTs (#1) and (#2), respectively. Consequently, the movement of these currents will generate magnetic fields surrounding CNTs (#1) and (#2) in accordance with Ampere’s circuital law within Maxwell’s equations, which connects the magnetic field encircling a closed loop to the electric current flowing through the loop.

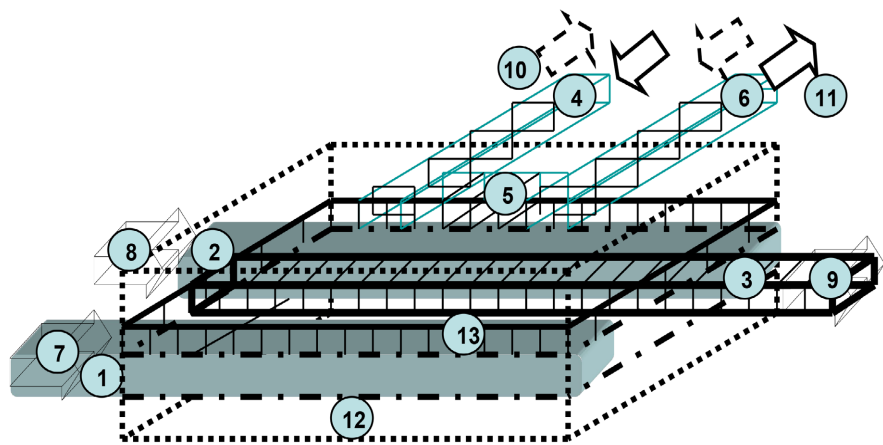


Figure 12. CNT-based magnetic multiplexer.

The operation of the device in **Figure 12** proceeds as follows: At first, CNT (#3) touches either CNT (#1) or CNT (#2), resulting in a current flowing through CNT (#3) in the same direction as currents (#7) and (#8). An electric current flows through CNTs (#4)-(#6), able to move in either the clockwise direction (#10) or the counterclockwise direction (#11). The electric current (#10) that passes through CNTs (#4)-(#6) will create a magnetic field in accordance with Maxwell’s equations, oriented per the right-hand thumb rule. If current (#11) travels through CNTs (#4)-(#6), it will produce a magnetic field in the opposite direction to that created by current (#10). The flow direction of the current in CNTs (#4)-(#6), and consequently the generated magnetic fields, acts as a control signal in a standard multiplexer; when current (#11) flows through CNTs (#4)-(#6), an attractive Lorentz force arises between the two current-carrying conductors, specifically between CNTs (#5) and (#3), causing CNT (#3) to move in space (#13) toward CNT (#5), resulting in contact with CNT (#2), indicating that current (#8) or input A is chosen to flow to the output. Conversely, if current (#10) is present in CNTs (#4)-(#6), a repulsive Lorentz force will be generated between CNT (#5) and CNT (#3), causing CNT (#3) to shift in the space (#13) away from CNT (#5) and subsequently create contact with CNT (#1), indicating that current (#7) or input B is

chosen to flow to the output. Therefore, the CNT-based device in **Figure 12** implements the operation of a two-to-one multiplexer.

4.4. CFE-Based Multiplexer

This Subsection presents important background on carbon field emission (CFE) and its utilization in multiplexing which includes both CNT-based field emitters and carbon fiber-based field emitters.

4.4.1. CNT-Based Field Emitters

Field emission refers to the release of electrons from a cathode's surface due to an applied electric field, which is contingent on the work function of the material that emits. This part highlights functional modeling of field emitters along with the relevant experimental configurations and assessments that define their time-dependent behaviors.

To conduct the static modeling of CNT-based field emitters, four field emission CNTs illustrated in **Figures 13(b)-(e)** were produced by Xintek, Inc. The copper anode is positioned on the right, while the utilized CNT emitter is fixed to a tungsten wire connected to the copper cylinder on the left, as depicted in **Figure 13(a)**. **Figures 13(b)-(e)** show the images of CNT emitters for each CNT captured using a JEOL Ltd. model JEM 6300 Scanning Electron Microscope (SEM). Carbon nanotubes M-1 and M-4 utilize one MWCNT as the emitter, while carbon nanotubes C-3 and C-6 employ one SWCNT as the emitter. The CNTs employed were bundled together with diameters ranging from 10 to 30 nm, though field emission from each CNT occurs primarily at the single CNT at the bundle's tip where the electric field is strongest.

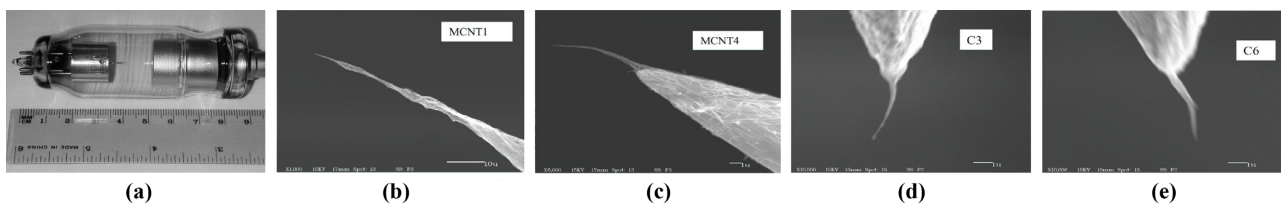


Figure 13. CNT-based field emission: (a) structure of the field emission CNTs, and (b)-(e) images from Scanning Electron Microscopy of the CNT-based emitters for the four utilized CNTs.

The DC current-voltage behaviors were assessed for these four CNTs alongside a field emitter tube from Leybold Didactic GmbH, featuring an etched single crystal of tungsten as the emitter. All measurements taken with the five tubes were conducted at room temperature. The tungsten tip is attached to a filament, allowing it to be heated for cleaning just before each measurement session. Nevertheless, cleaning the CNT is not feasible, which likely leads to the “switch-on” phenomenon where the supply voltage needs to be temporarily elevated significantly above the operating point to trigger field emission with the CNT. The data obtained from the DC measurements were processed using the Fowler-Nordheim analysis which relies on a simplified version of the Fowler-Nordheim equation

[23] [24] that determines the current density magnitude as a function of the applied static field for field emission from a certain material as follows:

$$J = AE^2 e^{(-B/E)} \quad (3)$$

where J and E are the magnitudes of the current density and the electric field intensity, and A and B are parameters that are functions of the work function Φ , where work function values are $\Phi = 4.5$ eV for tungsten and for the CNT was set to $\Phi = 4.9$ eV.

In order to apply the Fowler-Nordheim equation to the DC current-voltage data, the following equation was also used which is valid for a given CNT, where I is the field emission current and V is the potential applied between the anode and the cathode:

$$I = CV^2 e^{(-D/V)} \quad (4)$$

Equations (3) and (4) can be combined to obtain the following equations for the parameters S and R , which are used to characterize the field emitters:

$$S = CD^2 / AB^2 \quad (5)$$

$$R = V/E = D/B \quad (6)$$

where the parameter S denotes the effective emitting area, defined as the physical area of the emitter assuming a uniform current density across a specific area and zero in other regions, while the parameter R signifies the effective radius of curvature of the emitter, also accounting for the local enhancement of the electric field caused by the elongation of the emitter or the decline of the field resulting from shielding by neighboring structures.

The Fowler-Nordheim plots of the DC current-voltage information were created utilizing $\ln(I/V^2)$ as the y -axis and $(1/V)$ as the x -axis. **Figure 14** shows the unusual data for CNT C-6 that were gathered with the 2.575 G Ω ballast resistor. The effective radius of curvature values of the emitter R ranged from 77 to 110 nm for the four CNTs featuring CNT emitters. This indicates that the local electric field values at the emission sites reached as high as 14 V/nm in several of these measurements. It is shown that examining the field emission from different CNTs had estimated the electric field values by taking the applied voltage and dividing it by the distance from the anode to the emitting tip. The Fowler-Nordheim analysis indicated a value of 91 nm for parameter R in the Leybold tube, implying that the local electric field reached up to 5 V/nm in several of the measurements taken. Thus, the value of parameter R which was obtained for the Leybold tube appeared to be within reasonable range.

The Fowler-Nordheim analysis revealed that the effective emitting area S ranged from 81 to 230 nm² for the four CNTs with CNT emitters. If the existing density were uniform, this would relate to circular emitting areas with radii of about 5 - 9 nm. Lorenz microscopy has been utilized by others to directly identify the emission locations for field emission from MWCNT, revealing one or multiple sites with radii of a few nanometers, and their findings were in good alignment with other conducted experiments.

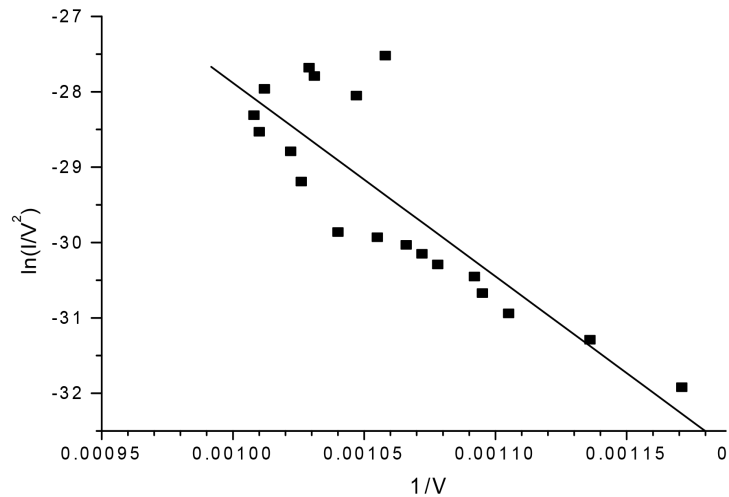


Figure 14. The resulting Fowler-Nordheim plot for CNT C-6.

4.4.2. Carbon Fiber Nanotips-Based Field Emitters

Because of the technological significance of carbon fibers, there has been an increasing interest in comprehending the mechanism of field electron emission from these fibers when subjected to an applied electric field. Employing carbon fibers as cathodes provides numerous advantages, including the capacity to function in a moderately high pressure, straightforward emitter production, excellent current consistency and extended longevity of the emitter. These cathodes are constructed from carbon fibers or other carbon-based substances. **Figure 15** displays a scanning electron image of a finely pointed carbon fiber tip at approximately 10,000× magnification.

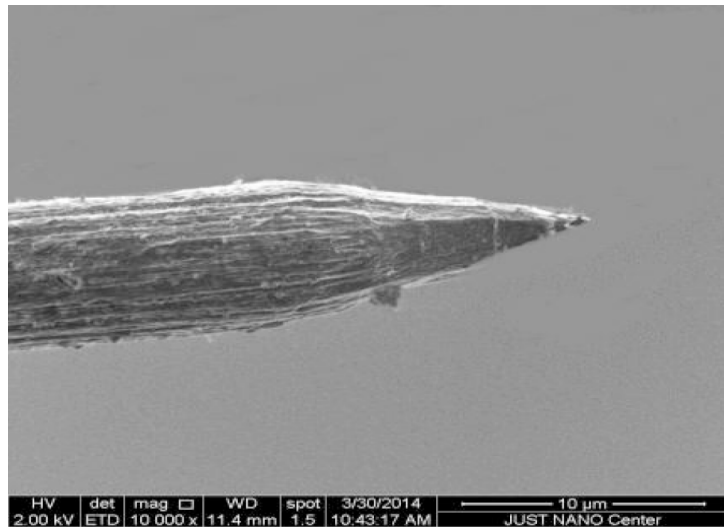


Figure 15. Scanning electron image of a very sharp carbon fiber nanotip.

To capture the emission characteristics, the voltage from extra high tension (EHT) to the tip is gradually increased until the emission current reached approximately one microampere on the picoammeter device, and then the voltage is gen-

tly lowered until the emission current disappears [25]. A linear Fowler-Nordheim plot is anticipated within this range. Throughout the experiments, a standard digital camera captured electronic emission images to examine the spatial distribution and consistency of the emission current. Both stability and brightness are crucial elements in assessing the quality of the electron source for real-world applications. A highly pointed carbon fiber tip was evaluated during the sample conditioning process with the apex radius of this tip measuring approximately 57 nm. **Figure 16** illustrates the emission properties obtained from the I-V characteristics along with the related Fowler-Nordheim plots.

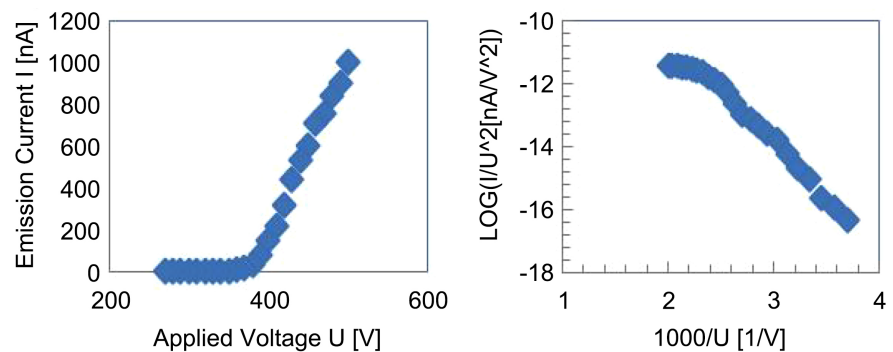


Figure 16. The I-V characteristics (left) and FN plot (right) of a very sharp carbon fiber nanotip during the cooling process, where the emission current stability becomes much higher.

4.4.3. CFE-Based Multiplexing

Through the application of the previously tested and noted characteristics and functions of CFE from the associated CNTs and nano-apex carbon fibers discussed in preceding Subsections 4.4.1 and 4.4.2, **Figure 17** illustrates the CFE-based primitive that accomplishes the two-to-one controlled switching.

In **Figure 17**, the control signal used for managing the device's electrical conductivity is realized through the applied electric field intensity (E), or alternatively, the work function (Φ) or voltage (V). The functioning of the device based on carbon field emission shown in **Figure 17(b)**, proceeds as follows: by applying a high voltage (HV) control signal, the voltage difference between the carbon cathodes and the opposing anode is altered. This adjustment will enable the carbon cathode receiving the control signal (HV) to exhibit field emission, while the other carbon cathode guided by the complementary control signal will not display field emission. When the voltage difference is inverted, the carbon cathode receiving the complementary control signal will exhibit field emission whereas the other carbon cathode controlled by the signal (HV) will not display field emission. Hence, this device carries out the operation of the 2-to-1 controlled-switching illustrated in **Figure 17(a)**. The experimental findings indicate that the spatial distance d which is needed between the cathodes and the opposing anode should be approximately 10 mm, otherwise beam distortion may happen impacting the current collected at the anode screen.

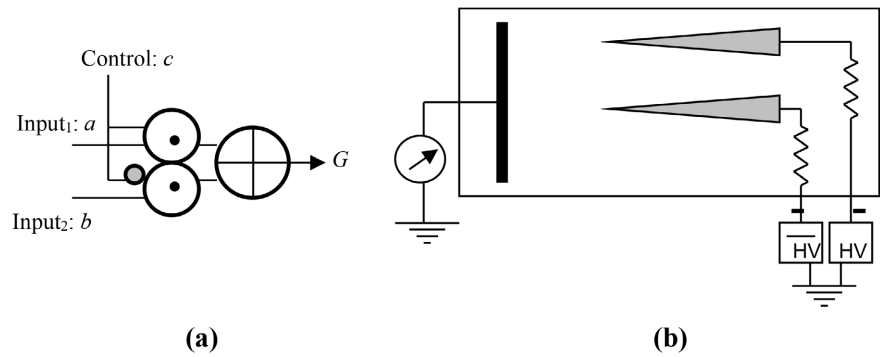


Figure 17. The CFE-based device implementing the operation of the two-to-one controlled-switching (CS): (a) two-to-one multiplexer ($G = ac + b\bar{c}$), and (b) carbon field emission-based two-to-one CS.

4.5. Three-Valued Processing via CNT-Based and CFE-Based Multiplexing

The previously presented background that includes CNT-based solid-state multiplexing in Subsection 4.2, CNT-based magnetic multiplexing in Subsection 4.3 and CFE-based multiplexing in Subsection 4.4, can be further utilized to conduct the important operation of many-valued controlled-switching. Synthesizing many-to-one controlled-switching can be achieved using the fundamental two-to-one controlled-switches from Subsections 4.2-4.4. For example, one needs two multiplexers to realize the functionality of three-to-one controlled-switching which is illustrated in **Figure 18** for three-valued logic.

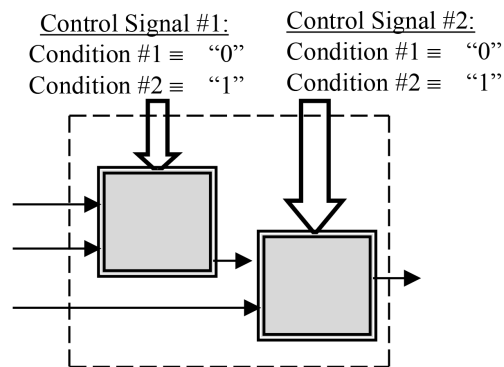


Figure 18. Carbon-based three-to-one controlled-switching, where the two multiplexing devices can be implemented using any of the controlled-switches from Subsections 4.2-4.4.

In **Figure 18**, the initial device D1 produces a signal from two input signals, while the subsequent device D2 generates a signal from other two input signals, thus enabling the device in **Figure 18** to function as a three-to-one multiplexer. In general, for m -valued logic, it is necessary to have $(m-1)$ of the two-to-one controlled switches from Subsections 4.2-4.4 to realize the function of m -to-1 multiplexing. The carbon-based implementation of Galois-based arithmetic functions

can be carried out as shown in **Figure 19**. In **Figure 19(b)**, the variables $\{A, B\}$ are ternary input variables with values $\in \{0, 1, 2\}$, and inputs C_k where $k \in \{0, 1, 2, 3\}$ serve as two-valued control variables that can take values $\in \{0, 1\}$. Observe that **Figure 19(b)** executes the related GF(3) addition and multiplication operations that were shown in **Table 1** by applying the suitable values of control variables C_k that choose the relevant device inputs $\{A, B\}$ and constant inputs $\{0, 1, 2\}$.

For example, **Table 2** illustrates a case for the execution of GF(3) addition and multiplication operations that were displayed in **Table 1** using **Figure 19(b)**. In **Table 2**, the symbol (+) indicates GF(3) addition, symbol (*) denotes GF(3) multiplication, $C_k (+)$ refers to the control variable for executing the ternary Galois-addition operation, while $C_k (*)$ signifies the control variable for performing the ternary Galois multiplication operation. Given that three-valued lattice grids will be constructed using the Galois addition and multiplication operations outlined in **Table 1**, the circuit illustrated in **Figure 19(b)** can be applied in three-valued configurations whenever third radix Galois operations are executed. As an illustration, any system-level hierarchical execution can be achieved through the repeated application of the carbon-based multiplexing device depicted in **Figure 19**, where the relevant arithmetic functions can be performed using the circuit in **Figure 19(b)** and by employing the pertinent input values listed in **Table 2**.

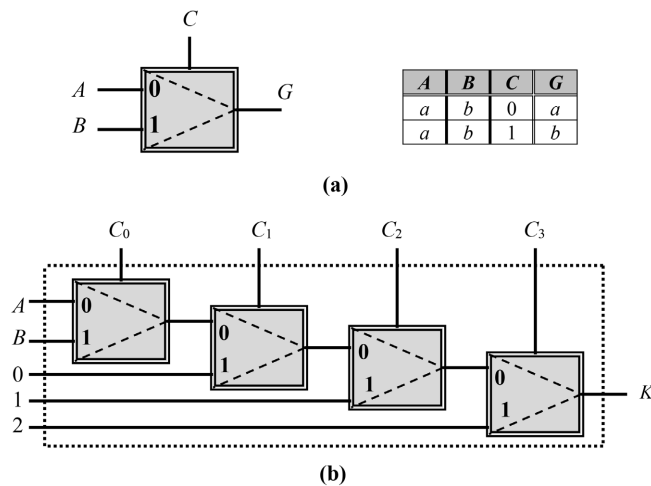


Figure 19. Carbon-based implementation of Galois arithmetic operations: (a) controlled-switch that can be implemented using any of the multiplexing devices in Subsections 4.2-4.4, and (b) circuit that uses controlled-switching to implement the corresponding GF(3) addition and multiplication operations from **Table 1**.

The creation of three-valued functions through three-dimensional carbon-multiplexed lattice grids is achievable, with the lattice's internal multiplexing nodes being implemented via CNT-based solid-state multiplexer (cf. Subsection 4.2), CNT-based magnetic multiplexer (cf. Subsection 4.3), or CFE-based multiplexer (cf. Subsection 4.4). As an instance of implementation, Example 3 illustrates the realization of a ternary non-symmetric function in a three-dimensional lattice grid by repeating variables. The operations conducted at each node in the associ-

ated three-dimensional lattice grid can be executed using the circuit depicted in **Figure 19(b)** along with the corresponding specified input values from **Table 2**.

Table 2. The implementation of GF(3) addition and multiplication operations from **Table 1** using **Figure 19(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

Example 3. For the three-valued non-symmetric function $F = ab + a'b''$ depicted in **Figure 20**, **Figure 21** demonstrates the implementation of this non-symmetric function in a three-dimensional regular lattice grid by employing variable repetition. In **Figure 21**, if one multiplies each leaf value, going counter clock wise, with all possible paths, from the leaves to the root, and adds them using the corresponding Galois operations from **Table 1** then one will obtain the corresponding required map in **Figure 20**, where $\{^0a, ^1a, ^2a\}$ are the zero, first and second polarities of the 1-RPL of variable a , $\{^0b, ^1b, ^2b\}$ are the zero, first and second polarities of the 1-RPL of variable b , and variables a and b can take any value $\in \{0, 1, 2\}$.

$ \begin{array}{c ccc} a \backslash b & 0 & 1 & 2 \\ \hline 0 & 2 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 2 & 0 & 2 & 1 \end{array} $				$ \begin{array}{c ccc} aa \backslash b & 0 & 1 & 2 \\ \hline 00 & 2 & 0 & 1 \\ 01 & - & - & - \\ 02 & - & - & - \\ 10 & - & - & - \\ 11 & 1 & 1 & 1 \\ 12 & - & - & - \\ 20 & - & - & - \\ 21 & - & - & - \\ 22 & 0 & 2 & 1 \end{array} $			
$F = ab + a'b''$							
(a)				(b)			

Figure 20. Two-variable non-symmetric three-valued function $F = ab + a'b''$: (a) map of three-valued function F which is non-symmetric where a' is a single shift to the value of variable a and b'' is a double shift to the value of variable b , and (b) repeating variable a to achieve logic symmetrization of function F .

The resulting three-dimensional lattice grid (**Figure 21**) possesses high regularity, which facilitates compactness in three-dimensional space, ease of manufacturability, efficient testability, and low-power consumption due to its use of local interconnects. Galois operations performed at each internal node can be implemented using the switching circuit shown in **Figure 19(b)** and utilizing the specific input values from **Table 2**. Furthermore, inner multiplexing can be achieved through the CNT-based solid-state and magnetic multiplexers presented in Subsections 4.2 and 4.3, as well as the CFE-based multiplexer utilizing CNTs and nano-apex carbon fibers described in Subsection 4.4.

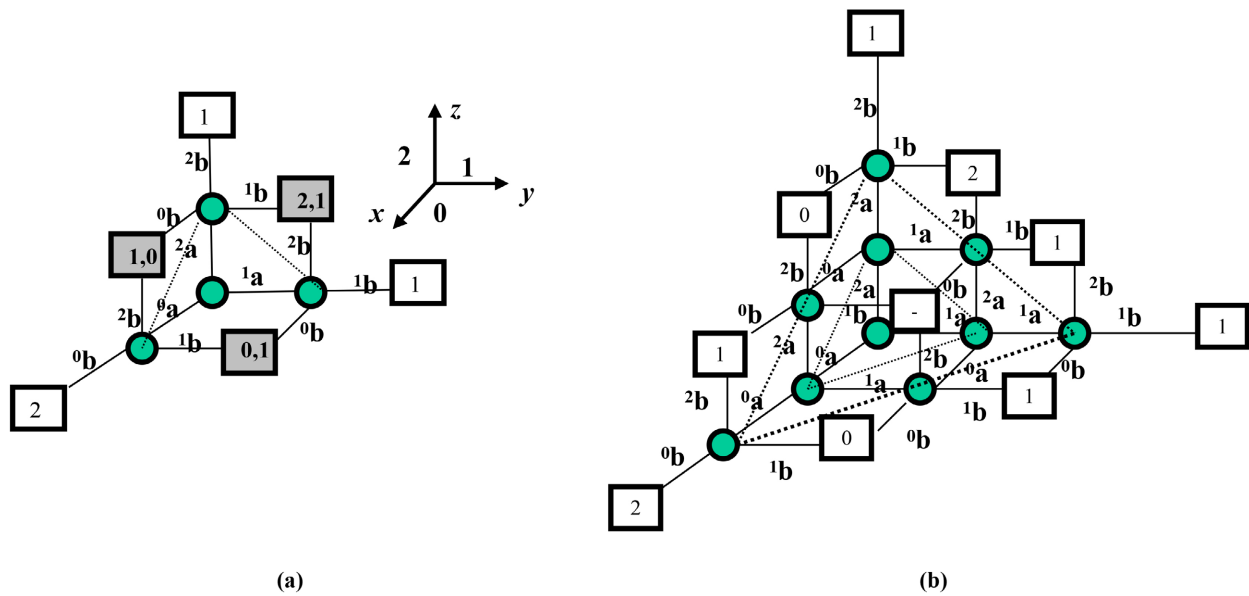


Figure 21. Regular nano-based three-dimensional lattice grids: (a) lattice grid corresponding to **Figure 20(a)** showing conflicting leaves in dark boxes, and (b) lattice grid corresponding to **Figure 20(b)** showing non-conflicting leaves achieved via variable repetition. The operations performed at each node in the resulting lattice grid can be implemented using the switching circuit from **Figure 19(b)** and the input values specified in **Table 2**.

5. Regular Nano-Based Lattice Circuit Design of the New Homeomorphic Joint Cybersecurity-Channel Coding Scheme

This Section introduces the nano-based regular lattice circuit synthesis of the homeomorphic joint Viterbi decoder-symmetric-key cybersecurity design, as an important implementation example of the new joint cybersecurity-channel coding scheme, which consists of the implementation of the two sequentially interconnected parts of (a) homeomorphic Viterbi decoder and (b) symmetric-key homeomorphic Algorithm_n. The internal functions within the new design are systematically implemented using regular lattice grids as was demonstrated in Section 3. The multiplexing nodes in the resulting lattice grids are then practically achieved utilizing the multiplexing operations of CNT-based solid-state and magnetic devices that were presented in Subsections 4.2 and 4.3, and CFE-based multiplexing from CNTs and nano-apex carbon fibers that were presented in Subsec-

tion 4.4. It is to be noted that the used representation of circuit notations, for the circuits that are used in this Section, follows closely the notation which is used within nano-scale quantum computing circuits and systems [8] [9]. While the design shown in this Section is implemented for the receiver side as an example (*i.e.*, homeomorphic joint Viterbi decoder-key-generated Algorithm_n), the same design method can be used for the circuit design implementation at the sender side as well (*i.e.*, homeomorphic joint key-generated Algorithm_n—convolutional encoder).

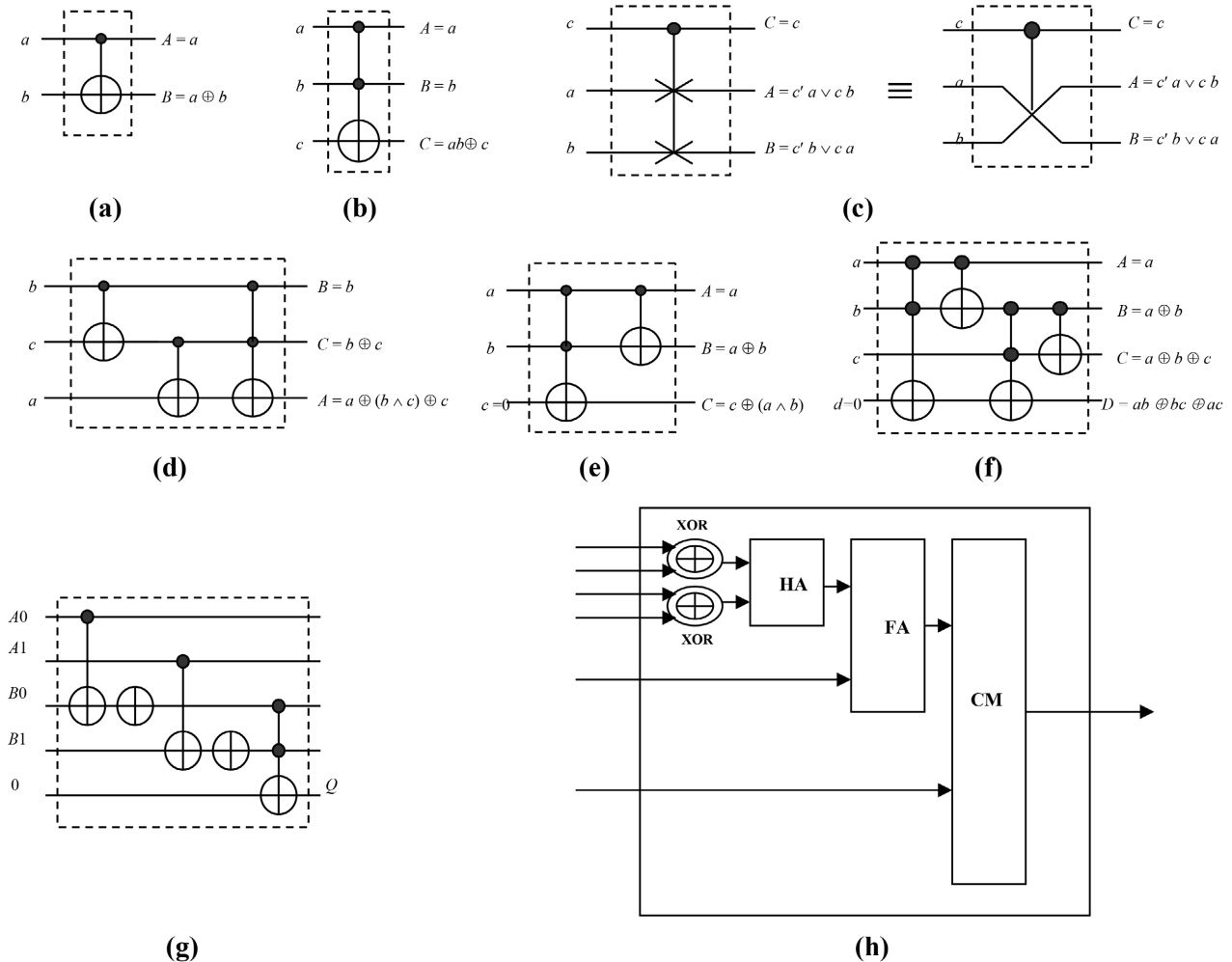


Figure 22. Logically homeomorphic circuits for the actualization of each cell in the homeomorphic Viterbi decoder.

The hardware implementation for each Viterbi cell in the (homeomorphic) Viterbi decoder requires the following homeomorphic components [17]: XOR, adder, subtractor and multiplexer to be paired in the synthesis of the homeomorphic comparator. **Figure 22** shows the various utilized circuits for the realization of each Viterbi cell in the corresponding (homeomorphic) Viterbi decoder [17], where **Figure 22(a)** shows the XOR gate (also known as Controlled-NOT gate), **Figure 22(b)** shows Controlled-Controlled-NOT gate, **Figure 22(c)** shows

multiplexer (also known as Controlled-Swap gate), **Figure 22(d)** shows subtractor, **Figure 22(e)** shows half-adder (HA), **Figure 22(f)** shows full-adder (FA), **Figure 22(g)** shows equality-based comparator (COMP) that compares two 2-bit numbers where an isolated XOR symbol means a NOT gate, and **Figure 22(h)** presents basic homeomorphic Viterbi cell which consists of two Controlled-NOT gates which are succeeded by three sequenced stages of HA, FA and COMP with multiplexing. As stated previously, the COMP can be acquired using a subtractor and a multiplexer.

A circuit for comparing two three-digit binary values, $X = [x_1, x_2, x_3]$ and $Y = [y_1, y_2, y_3]$, is synthesized in **Figure 23**. Using n -cells and one output circuit, the circuit in **Figure 23** can be expanded to compare two n -digit binary integers. A COMP circuit, including a COMP cell (**Figure 24(a)**) and a COMP output circuit (**Figure 24(b)**), is shown in **Figure 24**. A circuit for comparing two three-digit binary numbers is shown in **Figure 25(a)**, and the synthesis of a COMP with multiplexing is shown in **Figure 25(b)** [17]. As previously stated, by using n cells and one output circuit, the circuit in **Figure 25(a)** can be expanded to compare two n -digit binary integers.

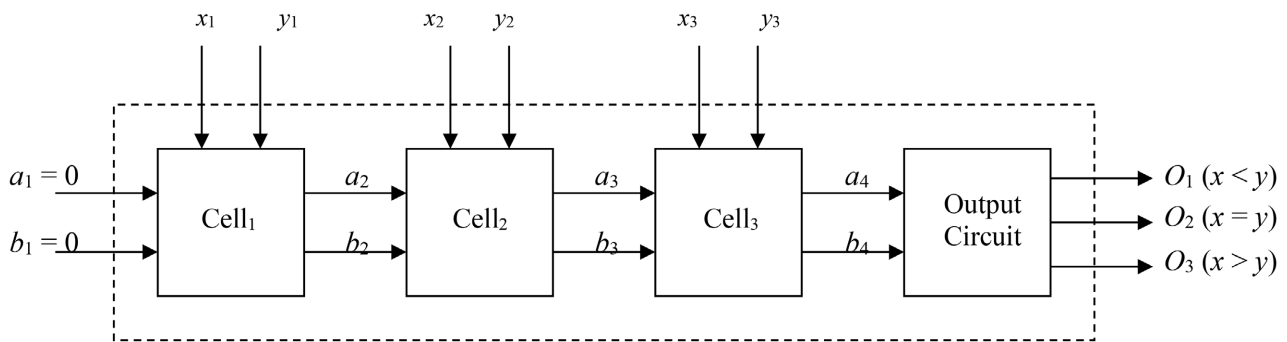


Figure 23. Comparing two 3-digit binary numbers $X = [x_1, x_2, x_3]$ and $Y = [y_1, y_2, y_3]$.

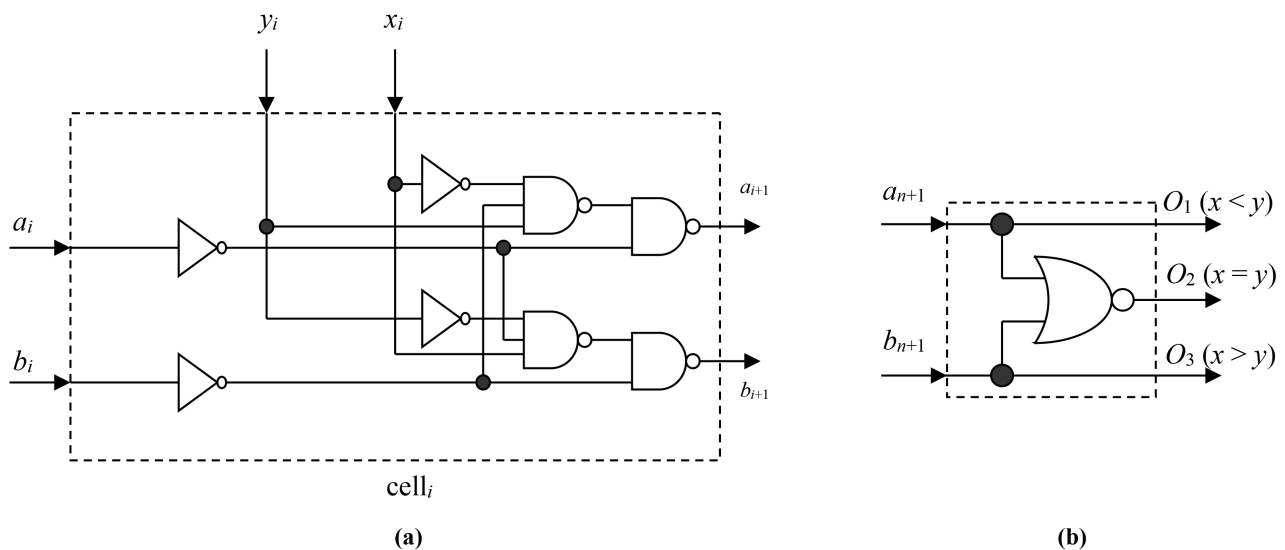


Figure 24. Synthesis of a Comparator: (a) cell and (b) output circuit.

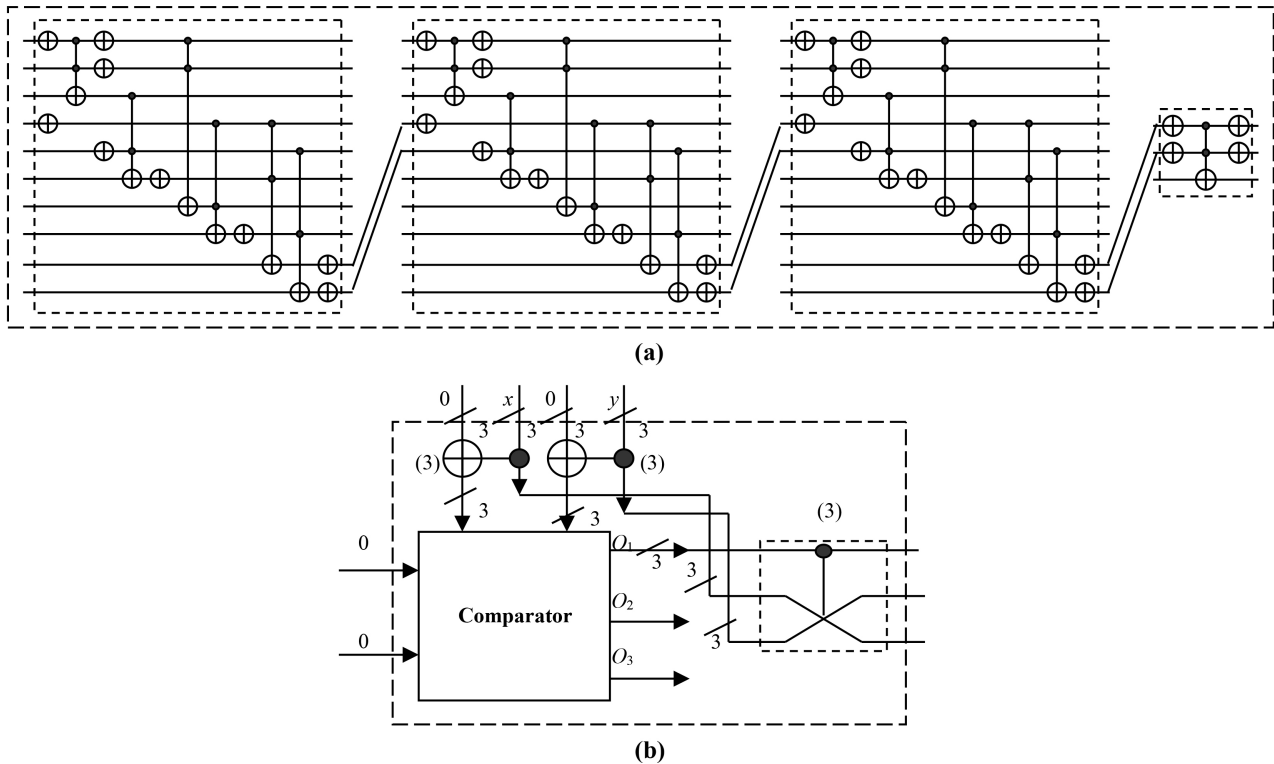


Figure 25. Synthesis of a COMP with multiplexing: (a) circuit for comparing two 3-digit binary numbers, and (b) COMP with multiplexing. Numbers 3 and (3) beside lines mean three copies.

Figure 26 presents the synthesis of a Viterbi cell in the Viterbi decoder that was shown in **Figure 22(h)** [17], where **Figure 26(a)** and **Figure 26(b)** show circuits which consist of two XORs to yield the difference between incoming received bits followed by HA to yield the corresponding sum which is the Hamming distance, **Figure 26(c)** shows a circuit which consists of HA and FA that adds the current Hamming distance to the previous Hamming distance, **Figure 26(d)** and **Figure 26(e)** show circuits that consist of HA followed by FA, and **Figure 26(f)** shows the corresponding COMP with multiplexing in the Viterbi cell for comparing the two arriving metric numbers $\{X = [s_3, s_4, c^*], Y = [s_3^*, s_4^*, c^{**}]\}$ selecting using control line O_1 the path with the least arriving metric (that is $X < Y$).

In the case when the paths that are going into a Viterbi cell are compared and their corresponding metrics are found to be the same, then a choice is made between the two paths, and this is performed in the circuit in **Figure 26(f)** since if $(\{s_3, s_4, c^*\} < \{s_3^*, s_4^*, c^{**}\})$ then $O_1 = "1"$ and thus it chooses X , else $O_1 = "0"$ and then it chooses Y for the other two larger or equal cases.

Each function inside the presented design of homeomorphic Viterbi decoder is implemented using a regular lattice grid as was shown in Section 3. This is done either by 1) implementing each function separately using a regular lattice grid and then interconnecting the resulting lattice grids to produce the total function or 2) producing the total function from sub-functions utilizing logic synthesis methods [26] [27] and then implementing the resulting total function at once using the

corresponding regular lattice grid. In either way, the multiplexing nodes within the resulting lattice grid(s) are then practically realized utilizing the CNT-based and CFE-based controlled-switching devices and methods that were presented and demonstrated in Section 4.

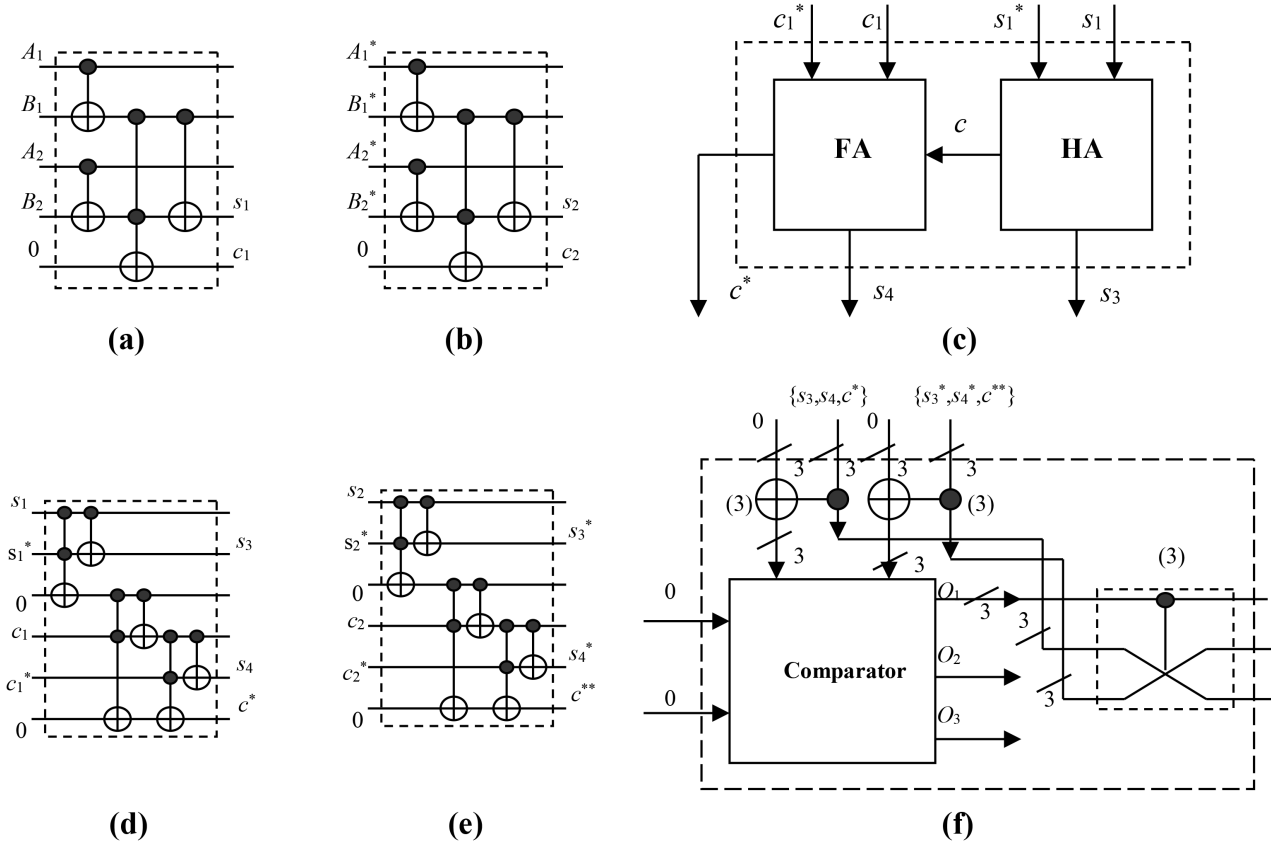


Figure 26. Synthesis of a Viterbi cell from **Figure 22(h)**.

Figure 27 presents the block diagram at the sender side for the introduced joint cybersecurity-channel coding using logic homeomorphism that was introduced for the first time in the first part of this article. This is achieved by encrypting the incoming data_{*j*} using homeomorphic mapping Algorithm_{*n*} into data_{*j*}. This homeomorphic mapping is done using a homeomorphic Algorithm_{*n*} which is generated from symmetric key *k_s*, which is shared between the sender and the receiver, where data_{*j*} can be decrypted at the receiver side into data_{*j*} iff the same symmetric key *k_s* has been used. After that, the encrypted data_{*j*} is fed into the next stage of channel encoding to produce the corresponding homeomorphic channel encoded data_{*k*}. At the receiver side, the reverse-order and inverse-function of this stage of joint cybersecurity-channel coding is performed using firstly homeomorphic channel decoding (e.g., using the homeomorphic Viterbi decoder) to produce data_{*j*} from received (erroneous) data_{*k*} and then using symmetric key *k_s* to generate the corresponding logically homeomorphic Algorithm_{*n*} to decrypt data_{*j*} into the primary data_{*i*}.

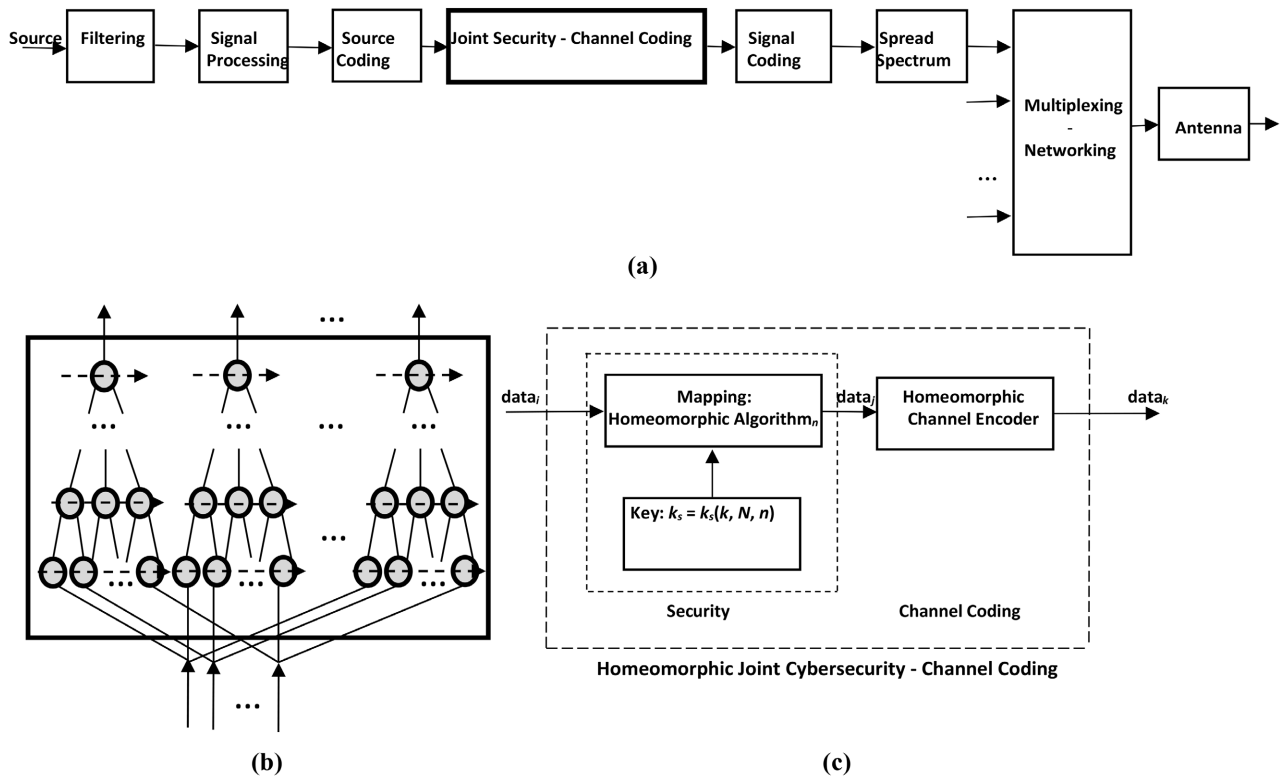


Figure 27. The structural block diagram of the new scheme of homeomorphic joint cybersecurity-channel coding at the sender side: (a) the new joint cybersecurity-channel coding scheme, (b) multi-input multi-output (MIMO) extension of lattice grids, such as the MIMO extensions of the 2D and 3D lattices from Section 3, where the resulting multi-output from this first stage is then fed into the second stage of homeomorphic channel encoding as shown in (c), and (c) block diagram of the sender two-stage joint cybersecurity-channel coding implementation. At the receiver side, the reverse-order and inverse-function of this joint cybersecurity-channel coding is performed using firstly homeomorphic channel decoding (e.g., using the homeomorphic Viterbi decoder) to produce $data_j$ from received (erroneous) $data_k$ and then using symmetric key k_s to generate the corresponding logically homeomorphic map Algorithm_{*n*} to decrypt $data_j$ into the original $data_i$.

As detailed previously in the first part of the article, the parameters for the symmetric key k_s are $k_s = k_s(k, N, n)$ where k is the number of batched inputs, N is the number of outputs in the bijective mapping, and n is the index of the used logically homeomorphic Algorithm_{*m*}, where $N \geq k$. The size of the key $|k_s|$ is given by $|k_s| = |k| + |N| + |n|$. The key $k_s(k, N, n)$ is generated using a random number generator (RNG) combined with a mathematical algorithm. The space from which the symmetric key k_s can be chosen is enormous depending on the combination of random choices via RNG for the symmetric key parameters $\{k, N, n\}$. The distribution of the shared symmetric key k_s to the sender and receiver can be done using mechanisms such as cryptographic handshake or the more advanced highly-secured modern method of quantum key distribution. The corresponding table for the utilized homeomorphic algorithms is stored at the sender and receiver sides with index n indicating the address of each homeomorphic Algorithm_{*n*} in the lookup table. The stored homeomorphic algorithms can have n variants, depending on the procedure for geometrically achieving logic homeomorphism via value-based space-partitioning that leads to spatial partitions of peculiar values, where

(a) several n procedures can be used to obtain the corresponding several n homeomorphic algorithms Algorithm_n or (b) alternatively a universal homeomorphic procedure Algorithm_u is stored to produce the corresponding homeomorphic mappings which can produce for specific internal set of parameters $\mathbf{p}$ the corresponding specific Algorithm_n where $n = f(\mathbf{p})$.

For data transmission, since homeomorphic mapping is performed at the first stage as shown in **Figure 27(c)**, one needs to implement firstly the multi-input multi-output (MIMO) homeomorphic Algorithm_m which is generated from the symmetric key k_s , and this can be performed as shown in **Figure 27(b)** using the MIMO extension of regular lattice grids, such as the MIMO extensions of the 2D and 3D lattices from Section 3. The internal nodes of the resulting MIMO lattice grid are then realized using the nano-based switching devices and methods that were presented in Section 4. The resulting multi-output mapping (**Figure 27(b)**) as the result from the first stage within **Figure 27(c)** is then fed into the second stage of homeomorphic channel encoding as shown in **Figure 27(c)** where the implementation of the second stage is performed using the corresponding nano-based regular lattice design of the homeomorphic channel encoder. For data reception, the reverse-order and inverse-function of this stage of joint cybersecurity-channel coding is performed using firstly channel decoding implementation via the homeomorphic Viterbi decoder and then the corresponding implementation of homeomorphic key-generated Algorithm_n .

The newly introduced homeomorphic joint cybersecurity-channel coding scheme can be done in software using general purpose computing, or can be performed for higher-speed lower-power and smaller-size in application-specific IC (ASIC) hardware using CNT-based regular lattice methods (that were shown in Sections 3 - 4) for the corresponding homeomorphic cybersecurity-channel coding design (such as the homeomorphic joint Viterbi decoder-key-generated Algorithm_n at the receiver side). Also, one may note that this new homeomorphism-based cybersecurity scheme can be implemented in sequence in addition to the previously implemented steps of other security methods, such as RSA and AES encryption, that were applied prior to the phase of this new homeomorphic joint cybersecurity-channel coding scheme, where this utilized implementation will give further confidentiality for the transacted data.

It is to be noted that one can use multi-stage cybersecurity by using several sequenced homeomorphic algorithms of $\{\text{Algorithm}_1, \text{Algorithm}_2, \dots, \text{Algorithm}_n\}$ that are key-generated by $\{k_{s1}, k_{s2}, \dots, k_{sn}\}$ where *each row-wise output* from each algorithm becomes an input to the next encrypting algorithm and thus parallel *encrypting algorithm tree* structure is generated to model the sequenced phases of consecutive encrypting algorithms, where the multi-stage decryption at the receiver is done in reverse-order and inverse-function of $\{\text{Algorithm}_m, \dots, \text{Algorithm}_2, \text{Algorithm}_1\}$ that are key-generated by $\{k_{sm}, \dots, k_{s2}, k_{s1}\}$. **Figure 28** shows an implementation of this parallel encrypting algorithm tree using tree structure of MIMO lattice grids.

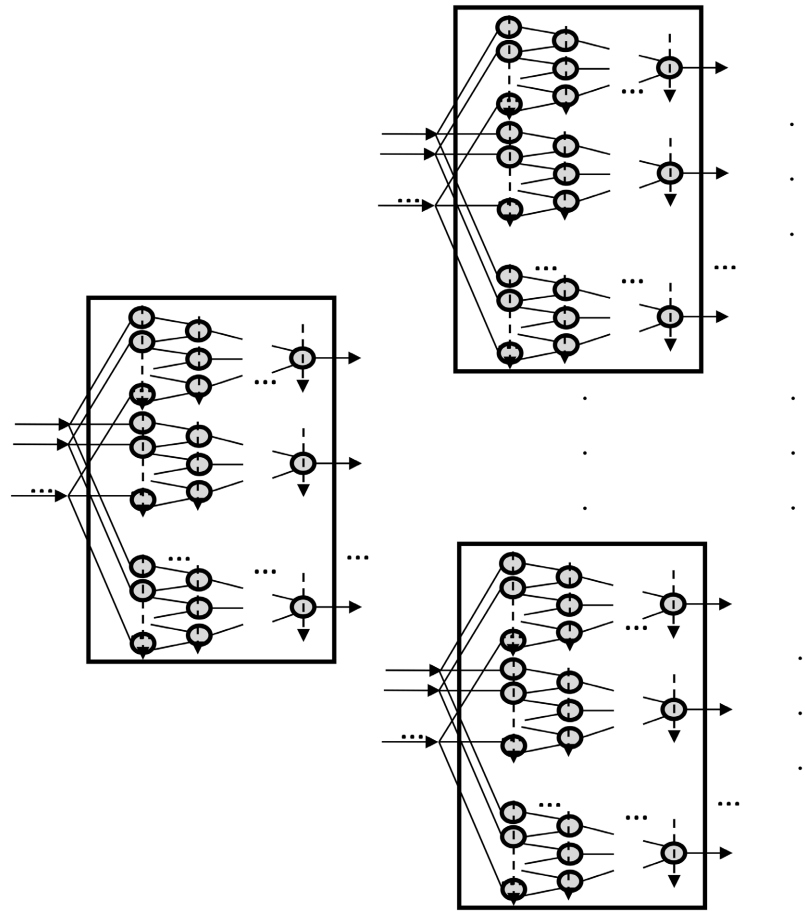


Figure 28. Tree of lattice grids to implement the corresponding multi-stage encryption at the sender by using several sequenced key-generated homeomorphic algorithms where each row-wise output from each stage becomes an input to the next stage and thus parallel encrypting tree structure is generated. The multi-stage decryption at the receiver is done using exact reverse-order and inverse-function steps.

The multiplexing nodes in the resulting parallel tree of MIMO lattice grids are then practically achieved utilizing the multiplexing operations of CNT-based solid-state and magnetic devices that were presented in Subsections 4.2 and 4.3, and CFE-based multiplexing from CNTs and nano-apex carbon fibers that were presented in Subsection 4.4. For the three-dimensional realization of such MIMO lattice grids, Galois operations that are performed in each internal node can be implemented using the switching circuit shown in **Figure 19(b)** and utilizing the corresponding specific input values from **Table 2**.

6. Conclusions and Future Work

This article introduces regular architectural implementation of the newly introduced homeomorphic joint cybersecurity-channel coding scheme, that was introduced for the first time in the first part of this article, using regular lattice grids and nano-based controlled-selectors, where multiplexing realization is performed using operations that utilize two-to-one controlled-switching elements that can

be directly implemented using the presented carbon-based multiplexing devices. In addition to the fact that logic homeomorphism is an essential requirement for the circuit synthesis within advanced technologies such as within nanoscale quantum computing circuits and systems, it was shown for the first time in the first part of this article that the property of logic homeomorphism can be useful to achieve higher confidentiality and data integrity (thus better reliability) enhancements within the new joint cybersecurity-channel coding scheme which are major characteristics within modern digital wireless networks and communication systems. Lattice grids possess the important property of high regularity which is very useful in low-power fault testing and localization, self-repair, compactness and ease of manufacturability. Further, because of only using local interconnects and the utilization of efficient carbon-based nano switching devices, the introduced nano-based lattice architecture can be utilized within several applications where speed, smaller size and minimal power consumption are required such as within contemporary ASIC design for more-reliable and better-secured modern wireless data networking and communications.

Future work will include the following items: 1) further comprehensive quantitative evaluation for security enhancements using the various design parameters within the newly introduced cybersecurity scheme, 2) evaluations for bit-error rate (BER), BER/SNR, total power consumption, time and area costs for typical constraint lengths, that will be performed in simulation and prototyping for the introduced hierarchical implementations of the new homeomorphic joint cybersecurity-channel coding scheme, 3) comparison with conventional counterparts within cybersecurity and channel coding methods for the various design parameters and using the corresponding standard benchmarks, 4) investigation of physical implementations using three-dimensional crystal lattice grids of the hierarchically synthesized three-dimensional lattice grids since cubical three-dimensional crystal lattices naturally exist where inter-related atoms existing in a potential field are spaced on the corners of three-dimensional cubes, and 5) investigation of further optimization algorithms for size reduction of the newly synthesized MIMO lattice grids using techniques such as variable re-ordering, choices of variable repetition and inner expansion nodes, 6) implementation using reconfigurable FPGA for the reprogrammable realization of the new symmetric-key homeomorphic Algorithm_m, and 7) the implementation of the newly introduced homeomorphic joint cybersecurity-channel coding scheme using Turbo codes.

Acknowledgements

The author Prof. Anas N. Al-Rabadi acknowledges a granted sabbatical leave in 2024-2025 from The University of Jordan that enabled him to achieve this extended research.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Proakis, J.G. and Salehi, M. (1994) Communication Systems Engineering. Prentice-Hall.
- [2] Lathi, B.P. (1995) Modern Digital and Analog Communication Systems. Oxford University Press.
- [3] Couch II, L.W. (2013) Digital and Analog Communication Systems. Pearson.
- [4] Al-Rabadi, A.N. (2009) Reversible Viterbi Algorithm and Its Closed-System Q-Domain Circuit Design and Computation. *Journal of Circuits, Systems and Computers*, **18**, 1627-1649. <https://doi.org/10.1142/s0218126609005903>
- [5] Bishop, M. (2005) Computer Security. Addison-Wesley.
- [6] Stallings, W. (2006) Cryptography and Network Security: Principles and Practices. Prentice Hall.
- [7] Forouzan, B. (2008) Cryptography and Network Security. McGraw-Hill.
- [8] Al-Rabadi, A.N. (2004) Reversible Logic Synthesis: From Fundamentals to Quantum Computing. Springer-Verlag.
- [9] Nielsen, M. and Chuang, I. (2000) Quantum Computation and Quantum Information. Cambridge University Press.
- [10] Al-Rabadi, A.N. (2009) Carbon NanoTube (CNT) Multiplexers, Circuits, and Actuators. Patent No. US 7,508,039 B. United States Patent and Trademark Office.
- [11] Landauer, R. (1961) Irreversibility and Heat Generation in the Computing Process. *IBM Journal of Research and Development*, **5**, 183-191. <https://doi.org/10.1147/rd.53.0183>
- [12] Al-Rabadi, A.N. (2005) Three-Dimensional Lattice Logic Circuits, Part I: Fundamentals. *Facta Universitatis Series: Electronics and Energetics*, **18**, 1-13. <https://doi.org/10.2298/fuee0501001a>
- [13] Al-Rabadi, A.N. (2005) Three-Dimensional Lattice Logic Circuits, Part II: Formal Methods. *Facta Universitatis Series: Electronics and Energetics*, **18**, 15-28. <https://doi.org/10.2298/fuee0501015a>
- [14] Al-Rabadi, A.N. (2005) Three-Dimensional Lattice Logic Circuits, Part III: Solving 3D Volume Congestion Problem. *Facta Universitatis Series: Electronics and Energetics*, **18**, 29-43. <https://doi.org/10.2298/fuee0501029a>
- [15] Al-Rabadi, A.N., Mousa, M.S., Al-Rabadi, R.F., Alnawasreh, S., Altrabsheh, R.F. and Madanat, M.A. (2016) Parallel Bijective Processing of Regular Nano Systolic Grids via Carbon Field Emission Controlled-Switching. *International Journal of Computer Science & Information Technology*, **8**, 77-101.
- [16] Al-Rabadi, A.N. (2020) Concurrency within Ternary Galois Processing of Highly-Regular 3D Networks via Controlled Nano Switching. *International Journal of Computer Science and Information Technology*, **12**, 1-23. <https://doi.org/10.5121/ijcsit.2020.12101>
- [17] Al-Rabadi, A.N. (2025) Convolutional Homeomorphic Error-Correcting Codes Utilizing Nano Controlled-Selector Architectural Effectuation. *Journal of Computer and Communications*, **13**, 187-226. <https://doi.org/10.4236/jcc.2025.1310010>
- [18] Smith, K. (2002) Prospects for VLSI Technologies in MVL. Ultra Large Scale Integration Workshop, Boston, MA.
- [19] Amaratunga, G. (2003) Watching the Nanotube. *IEEE Spectrum*, **40**, 28-32. <https://doi.org/10.1109/mspec.2003.1228005>

- [20] Bonard, J.M., Dean, K.A., Coll, B.F. and Klinke, C. (2002) Field Emission of Individual Carbon Nanotubes in the Scanning Electron Microscope. *Physical Review Letters*, **89**, Article 197602. <https://doi.org/10.1103/physrevlett.89.197602>
- [21] Collins, P.G. and Avouris, P. (2000) Nanotubes for Electronics. *Scientific American*, **283**, 62-69. <https://doi.org/10.1038/scientificamerican1200-62>
- [22] Derycke, V., Martel, R., Appenzeller, J. and Avouris, P. (2001) Carbon Nanotube Inter- and Intramolecular Logic Gates. *Nano Letters*, **1**, 453-456. <https://doi.org/10.1021/nl015606f>
- [23] Forbes, R.G. (2012) Extraction of Emission Parameters for Large-Area Field Emitters, Using a Technically Complete Fowler-Nordheim-Type Equation. *Nanotechnology*, **23**, Article 095706. <https://doi.org/10.1088/0957-4484/23/9/095706>
- [24] Fowler, R.H. and Nordheim, L.W. (1928) Electron Emission in Intense Electric Fields. *Proceedings of the Royal Society of London, Series A, Containing Papers of a Mathematical and Physical Character*, **119**, 173-181. <https://doi.org/10.1098/rspa.1928.0091>
- [25] Alnawasreh, S., Mousa, M.S. and Al-Rabadi, A.N. (2015) Investigating the Effects of Sample Conditioning on Nano-Apex Carbon Fiber Tips for Efficient Field Electron Emission. *Jordan Journal of Physics*, **8**, 51-57.
- [26] Katz, R. (1994) Contemporary Logic Design. Benjamin/Cummings Publishing.
- [27] Roth, C. (1995) Fundamentals of Logic Design. PWS Publishing.