Journal of Engineering and Computer Sciences
Qassim University, Vol. 8, No. 1, pp. 43-62 (January 2015/Rabi' I 1436H)

A Tutorial Exposition of the Analysis of Synchronous Boolean Networks via Semi-Tensor Products

Ali Muhammad Ali Rushdi and Fares Ahmad Muhammad Ghaleb

Department of Electrical and Computer Engineering,

Faculty of Engineering, King Abdulaziz University,

P. O. Box 80204, Jeddah, 21589, Saudi Arabia

arushdi@kau.edu.sa

(Received 13/9/2014; Accepted for publication 25/3/2015)

ABSTRACT. This paper is a detailed tutorial exposition of the analysis of synchronous Boolean networks via a particular matrix product called the Semi-Tensor Product (STP) of matrices, which multiplies two matrices $\boldsymbol{A}_{mn}$ and $\boldsymbol{B}_{pq}$ in which the column dimension n of the first matrix is not necessarily equal to the row dimension p of the second matrix, but is possibly a multiple or divisor of it. The state space of a Boolean network of n nodes is denoted herein by a vector of 2^n states in natural order obtained as the STP of n 2-element vectors representing the network variables. A notable contribution of the paper is that its matrix expression of logic follows the conventional truth-table order, and not the reverse unfamiliar order followed so far by the STP community. We reproduce the STP analysis of a classical example network**.** We include minute details that make the STP manipulations easily accessible to and more understandable by their potential users**.** Our analysis points to more efficient implementations of the STP solution in which the OR and XOR binary operations do not inflict a cost more than they really deserve.

Keywords: Synchronous Boolean networks, Semi-tensor products, Quasi-commutativity, Size reduction, Transition matrix.

1. Introduction

A synchronous Boolean network model is the simplest possible conceptual model that mimics or captures the essential features of many biological systems. Hence, this model became a powerful tool for describing, analyzing, and simulating these systems. The model consists of a set of n nodes, each of which is either in state 1 (On) or state 0 (Off) at any given time; t. Each node is updated at time $(t + 1)$ by inputs from any fixed subset of the set of nodes according to any desired logical rule [1-7].

Recently, the dynamics of synchronous Boolean networks have been extensively studied by a novel matrix method utilizing a new matrix product, called the Semi-Tensor Product (STP) of matrices [8-18]. Based on the STP paradigm, a certain matrix expression of logic is used for the numerical derivation of the transition matrix [T] of the Boolean network, which is called the structure matrix in [11]. This matrix is then analyzed to deduce full information about the transient and cyclic behavior of the network.

Despite the great success of the STP methodology, and despite the availability of many reviews on it [11, 19-22], it does not seem to be assimilated rapidly or fully enough by the scientific community, possibly due to its intrinsic difficulty. Since the usefulness of the STP methodology definitely outweighs its difficulty, no effort should be spared to make it easily accessible to researchers who need to utilize it. It is in this spirit that we have written our extensive survey [22], and that we write this current paper to popularize this long-awaited novel methodology.

While our earlier paper [22] is a tutorial exposition of the STP methodology with a stress on its somewhat *dubious* general representation of Boolean functions, this current paper deals with the *definitely successful* application of the STP methodology to *synchronous Boolean networks*. In fact, it is a detailed tutorial exposition of the analysis of synchronous Boolean networks via the STP methodology. We employ a particular semi-tensor product of matrices, which multiplies two matrices $\boldsymbol{A}_{mn}$ and $\boldsymbol{B}_{pq}$ in which the column dimension n of the first matrix is a multiple or divisor of the row dimension p of the second matrix. This means that we do not require the STP methodology in full (we do not demand that n be totally unrelated to p), but we utilize a simple special-case that represents a minimal departure from the conventional wisdom. We review basic definitions and operations of this particular STP, which particular emphasis on its use in swapping vectors or matrices and in reducing the size of a matrix. We represent a switching variable as a column vector of two elements, namely the complement of variable and the variable itself. This representation is in contrast to the one common in STP literature in which the order of these two elements is reversed. The state space of a Boolean network of n nodes, therefore, is a vector of 2^n states in *natural* order obtained as the STP of the n 2-element vectors representing the network variables. We reproduce the STP analysis of a classical example network, making sure to explain every minute detail and to expose every partial result. Through this detailed example, we hope to have made the STP manipulations easily accessible to and more understandable by a larger group of readers. We hope also to circumvent serious reservations against the STP application to problems involving Boolean functions. One reservation is that an STP representation of a Boolean function keeps records of both the function and its complement, and hence involves *unnecessary duplication* [22]. While this reservation is applicable in general logic problems, it vanishes in the case of the study of the state space of a Boolean network. In fact, the 2^n states for n variables reduce to 2 states for a single variable, and representing a variable by a 2-element vector becomes acceptable (even a necessity) in this case. Another reservation is that the STP solution uses the very general and *costly operation* of real multiplication to implement binary operations on 1-bit operands, such as the OR and XOR operations of Boolean algebra [22]. Our analysis herein points to *more efficient* implementations of the STP solution in which these binary operations do not inflict a cost more than they really deserve.

The rest of this paper is organized as follows. Section 2 reviews and explains the basic definitions of the tensor product [22-25], the semi-tensor product [11, 19-22], the swap matrix [11, 19], matrix expression of logic [11, 22, 26-30], and the power reducing matrix [11]. A novel contribution of the paper is that its matrix expression of logic follows the *conventional truth-table order*, and not the reverse order followed so far by the STP community. Section 3 presents a detailed STP solution of a small synchronous Boolean network, which culminates in the production of the correct transition matrix of the network. Section 4 concludes the paper.

2. Basic Definitions

2.1 The Kronecker (Tensor) product (KP):

The Kronecker (tensor) product is defined for any two matrices regardless of their dimensions [22-25]. Let $\boldsymbol{A}_{m\times n}=\{a_{ij}\}$, $(1 \le i \le m,\ 1 \le j \le n)$ be a matrix of m rows and *n* columns and $\boldsymbol{B}_{p\times q}$ be a matrix of *p* rows and *q* columns. The Kronecker product (KP) of $\boldsymbol{A}$ and $\boldsymbol{B}$ is denoted by the operator "$\otimes$", and is defined as the matrix $(\boldsymbol{A}\otimes\boldsymbol{B}) = \{a_{ij}\boldsymbol{B}\}$ obtained by multiplying each element of $\boldsymbol{A}$ by the matrix $\boldsymbol{B}$, namely

$$(\boldsymbol{A}\otimes\boldsymbol{B})_{mp\times nq} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \dots & a_{1n}\boldsymbol{B} \\ a_{21}\boldsymbol{B} & a_{22}\boldsymbol{B} & \dots & a_{2n}\boldsymbol{B} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \dots & a_{mn}\boldsymbol{B} \end{bmatrix}.$$

2.2 The Semi-Tensor Product (STP)

The left STP of two general matrices (usually called STP) is denoted by the $'\ltimes'$ operator, and given by the conventional matrix product of the Kronecker products of the respective matrices by appropriate unit matrices, namely

$$\boldsymbol{S}_{m\frac{c}{n}\times q\frac{c}{p}} = \boldsymbol{A}_{\mathrm{m\times n}} \ltimes \boldsymbol{B}_{\mathrm{p\times q}} = (\boldsymbol{A}_{\mathrm{m\times n}} \otimes \boldsymbol{I}_{\frac{c}{n}})_{\mathrm{m}\frac{c}{n}\times \mathrm{c}}(\boldsymbol{B}_{\mathrm{p\times q}} \otimes \boldsymbol{I}_{\frac{c}{p}})_{\mathrm{c\times q}\frac{c}{p}}, \tag{1}$$

where the integer c is the least common multiple (*lcm*) of the (totally unrelated) integers n and p, while I_k is the $k \times k$ unit (identity) matrix. The STP inherits many properties of the conventional matrix product, and reduces to it when $n = p$. In particular, the STP is *associative*, i.e.

$$\boldsymbol{A} \ltimes (\boldsymbol{B} \ltimes \boldsymbol{C}) = (\boldsymbol{A} \ltimes \boldsymbol{B}) \ltimes \boldsymbol{C} \tag{2}$$

The STP of two column vectors $\boldsymbol{A}_{\mathrm{m\times 1}}$ and $\boldsymbol{B}_{\mathrm{p\times 1}}$ is the column vector $\boldsymbol{S}_{mp\times 1}$ given by

$$\begin{aligned}\boldsymbol{A}_{\mathrm{m\times 1}} \ltimes \boldsymbol{B}_{\mathrm{p\times 1}} &= (\boldsymbol{A}_{\mathrm{m\times 1}} \otimes \boldsymbol{I}_{\mathrm{p}})(\boldsymbol{B}_{\mathrm{p\times 1}} \otimes \boldsymbol{I}_{1}) = (\boldsymbol{A}_{\mathrm{m\times 1}} \otimes \boldsymbol{I}_{\mathrm{p}})\, \boldsymbol{B}_{\mathrm{p\times 1}} = \boldsymbol{A}_{\mathrm{m\times 1}} \otimes \boldsymbol{B}_{\mathrm{p\times 1}} \\ &= [\mathrm{a_1b_1} \;\; \mathrm{a_1b_2} \;\; \cdots \;\; \mathrm{a_1b_p} \;\; \mathrm{a_2b_1} \;\; \mathrm{a_2b_2} \;\; \cdots \;\; \mathrm{a_2b_p} \;\; \cdots \;\; \mathrm{a_mb_1} \;\; \mathrm{a_mb_2} \;\; \cdots \;\; \mathrm{a_mb_p}]^{\mathrm{T}}.\end{aligned} \tag{3}$$

Important special cases of (1) are obtained when n is a multiple of $p (n = \ell p)$ or a divisor of it $(p = \ell n)$. For the case $n = \ell p$, equation (1) reduces to

$$\boldsymbol{S}_{\mathrm{m}\times \ell\mathrm{q}} = \boldsymbol{A}_{\mathrm{m\times n}} \; (\boldsymbol{B}_{\mathrm{p\times q}} \otimes \boldsymbol{I}_{\ell}), \tag{4}$$

while for the case $p = \ell n$, equation (1) reduces to

$$\boldsymbol{S}_{\ell\mathrm{m\times q}} = (\boldsymbol{A}_{\mathrm{m\times n}} \otimes \boldsymbol{I}_{\ell}) \; \boldsymbol{B}_{\mathrm{p\times q}}. \tag{5}$$

2.3 The Swap Matrix

Conventional matrix multiplication (and consequently STP) is *not commutative* in general. However, STP acquires some *quasi-commutative* properties with the aid of auxiliary tools called swap matrices [11]. We restrict our attention here to a particular *square swap matrix* $\boldsymbol{W}_{[\mathrm{m,p}]}$ of dimensions $mp \times mp$ that enforces the quasi commutative relation

$$\boldsymbol{B}_{\mathrm{p\times 1}} \ltimes \boldsymbol{A}_{\mathrm{m\times 1}} = \boldsymbol{W}_{[\mathrm{m\times p}]} \; \boldsymbol{A}_{\mathrm{m\times 1}} \ltimes \boldsymbol{B}_{\mathrm{p\times 1}}, \tag{6}$$

where $(\boldsymbol{A}_{\mathrm{m}\times 1} \ltimes \boldsymbol{B}_{\mathrm{p}\times 1})$ is given by (3), while $(\boldsymbol{B}_{\mathrm{p}\times 1} \ltimes \boldsymbol{A}_{\mathrm{m}\times 1})$ is given by

$$\boldsymbol{B}_{\mathrm{p}\times 1} \ltimes \boldsymbol{A}_{\mathrm{m}\times 1} = \boldsymbol{B}_{\mathrm{p}\times 1} \otimes \boldsymbol{A}_{\mathrm{m}\times 1} \tag{7}$$
$$= [b_1a_1 \;\; b_1a_2 \;\; \cdots \;\; b_1a_m \;\; b_2a_1 \;\; b_2a_2 \;\; \cdots \;\; b_2a_m \;\; \cdots \;\; b_pa_1 \;\; b_pa_2 \;\; \cdots \;\; b_pa_m]^T.$$

For the case $m = p = 2$, we have

$$\boldsymbol{B}_{2\times 1} \ltimes \boldsymbol{A}_{2\times 1} = \begin{bmatrix} b_1a_1 \\ b_1a_2 \\ b_2a_1 \\ b_2a_2 \end{bmatrix} = \boldsymbol{W}_{[2,2]}\, \boldsymbol{A}_{2\times 1} \ltimes \boldsymbol{B}_{2\times 1} = \begin{array}{c} [a_1b_1 \quad a_1b_2 \quad a_2b_1 \quad a_2b_2] \\ \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{bmatrix} \end{array} \tag{8}$$

For convenience, the vector $(\boldsymbol{A}_{2\times 1} \ltimes \boldsymbol{B}_{2\times 1})$ is not placed to the right of $\boldsymbol{W}_{[2,2]}$, but instead, its transpose is placed above it. This is a well-known trick used frequently to enhance the readability of conventional matrix multiplication [31].

For the case $m = 2$, and $p = 4$, we have

$$\boldsymbol{B}_{4\times 1} \ltimes \boldsymbol{A}_{2\times 1} = \begin{bmatrix} b_1a_1 \\ b_1a_2 \\ b_2a_1 \\ b_2a_2 \\ b_3a_1 \\ b_3a_2 \\ b_4a_1 \\ b_4a_2 \end{bmatrix} = \boldsymbol{W}_{[2,4]}\, \boldsymbol{A}_{2\times 1} \ltimes \boldsymbol{B}_{4\times 1} = \begin{array}{c} [a_1b_1 \quad a_1b_2 \quad a_1b_3 \quad a_1b_4 \quad a_2b_1 \quad a_2b_2 \quad a_2b_3 \quad a_2b_4] \\ \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \end{bmatrix} \end{array} \tag{9}$$

2.4 Matrix Expression of Logic in Conventional Order

A logic variable (A switching variable or a two-valued Boolean variable) is represented in the STP literature (see, e.g., [11, 22, 26-30]) as a 2×1 matrix of components x_i and $\overline{x}_i$, respectively. We will reverse these two components herein, to have

$$x_i \sim \boldsymbol{x}_i = \begin{bmatrix} \overline{x}_i \\ x_i \end{bmatrix}. \tag{10}$$

Corresponding to the definition (10), the Boolean constants 0 and 1 are given by

$$0 \sim \boldsymbol{\delta}_2^1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \tag{11}$$

$$1 \sim \boldsymbol{\delta}_2^2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{12}$$

Here, the notation $\boldsymbol{\delta}_n^j$ denotes the jth column of the $n \times n$ identity matrix $\boldsymbol{I}_n$. Boolean operators are defined as matrices, called structure matrices. The unary NOT operation ($\neg$) is expressed as

$$\boldsymbol{M}_\neg = \boldsymbol{\delta}_2[2 \quad 1] = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{13}$$

so that

$$\boldsymbol{M}_\neg \ltimes \boldsymbol{x}_{\mathrm{i}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \ltimes \begin{bmatrix} \overline{x}_i \\ x_i \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \overline{x}_i \\ x_i \end{bmatrix} = \begin{bmatrix} x_{\mathrm{i}} \\ \overline{x}_{\mathrm{i}} \end{bmatrix} = \overline{\boldsymbol{x}}_{\mathrm{i}}. \tag{14}$$

Before introducing binary operators, we must see first how a set $\boldsymbol{X}$ of two variables x_1 and x_2 are expressed:

$$\boldsymbol{X} = \begin{bmatrix} \overline{x}_1 \\ x_1 \end{bmatrix} \ltimes \begin{bmatrix} \overline{x}_2 \\ x_2 \end{bmatrix} = \left(\begin{bmatrix} \overline{x}_1 \\ x_1 \end{bmatrix} \otimes \boldsymbol{I}_2 \right) \left(\begin{bmatrix} \overline{x}_2 \\ x_2 \end{bmatrix} \otimes \boldsymbol{I}_1 \right) = \begin{bmatrix} \overline{x}_1 \boldsymbol{I}_2 \\ x_1 \boldsymbol{I}_2 \end{bmatrix} \begin{bmatrix} \overline{x}_2 \boldsymbol{I}_1 \\ x_2 \boldsymbol{I}_1 \end{bmatrix} = \tag{15}$$

$$\begin{bmatrix} \overline{x}_1 & 0 \\ 0 & \overline{x}_1 \\ x_1 & 0 \\ 0 & x_1 \end{bmatrix} \begin{bmatrix} \overline{x}_2 \\ x_2 \end{bmatrix} = \begin{bmatrix} \overline{x}_1 \overline{x}_2 \\ \overline{x}_1 x_2 \\ x_1 \overline{x}_2 \\ x_1 x_2 \end{bmatrix}.$$

The vector $\boldsymbol{X}$ constitutes the basis on which binary operators are constructed. As a result of the choice made in (10), it appears in the same order of what is used conventionally in truth tables in classical textbooks on logic design or Boolean algebra (see, e.g, [32]). The STP community typically produce $\boldsymbol{X}$ in an *unconventional reversed* order.

Table (1) summarizes (conventional-order) matrix representations of the six commutative binary Boolean operators in addition to the IMPLY operator. The (+) sign in Table (1) denotes the standard operation of real addition, but can be safely understood herein to express the *XOR* or even the *OR* Boolean operation. The STP representation of a n-variable switching function f takes the form of a 2×1 matrix of the form

$$\boldsymbol{M}_f \ltimes \boldsymbol{X} = \boldsymbol{M}_f \ltimes \boldsymbol{x}_1 \ltimes \boldsymbol{x}_2 \ltimes \cdots \ltimes \boldsymbol{x}_n = \boldsymbol{M}_f \boldsymbol{X} = \boldsymbol{M}_f(\boldsymbol{x}_1 \ltimes \boldsymbol{x}_2 \ltimes \cdots \ltimes \boldsymbol{x}_n), \quad (16)$$

where $\boldsymbol{M_f}$ is a 2×2^nbinary matrix whose rows are the truth tables of $\overline{f}$and f, respectively, and $\boldsymbol{X}$ is a $2^{\mathrm{n}} \times 1$ vector of the minterms over the variables $x_1, x_2, \cdots, x_n$. For example, for n = 3, the vector $\boldsymbol{X}$ is given by

$$\boldsymbol{X} = \boldsymbol{x}_1 \ltimes \boldsymbol{x}_2 \ltimes \boldsymbol{x}_3 = \\ [\overline{x}_1\overline{x}_2\overline{x}_3 \quad \overline{x}_1\overline{x}_2x_3 \quad \overline{x}_1x_2\overline{x}_3 \quad \overline{x}_1x_2x_3 \quad x_1\overline{x}_2\overline{x}_3 \quad x_1\overline{x}_2x_3 \quad x_1x_2\overline{x}_3 \quad x_1x_2x_3]^T. (17)$$

2.5 The power reducing matrix

The power reducing matrix $\boldsymbol{M}_r$ [11, p. 56-57] is a tool to invoke the idempotency of the AND operator $(x \wedge x = x)$, via

$$\boldsymbol{x} \ltimes \boldsymbol{x} = M_r\, \boldsymbol{x}, \quad (18)$$

$$\boldsymbol{M}_r = \boldsymbol{\delta}_4\,[1 \quad 4] = \begin{bmatrix} \mathbf{1} & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & \mathbf{1} \end{bmatrix}, \quad (19)$$

where both sides of (18) are equal to $[\overline{x} \quad 0 \quad 0 \quad x]^T$.

3. A detailed Example

We give herein a detailed explanation of example 5.7 in p.118 of [11]. This example deals with a 3-node synchronous Boolean network that has been analyzed also in [2, 5], and is governed by the equations:

$$x_1(t+1) = x_2(t) \wedge x_3(t), \tag{20a}$$

$$x_2(t+1) = 1 \oplus x_1(t), \tag{20b}$$

$$x_3(t+1) = x_2(t). \tag{20c}$$

In matrix form, these equations take the form

$$\boldsymbol{x}_1(t+1) = \boldsymbol{M}_\wedge \ltimes \boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t), \tag{21a}$$

$$\boldsymbol{x}_2(t+1) = \boldsymbol{M}_\oplus \ltimes \boldsymbol{\delta}_2^2 \ltimes \boldsymbol{x}_1(t) = \boldsymbol{M}_\neg \ltimes \boldsymbol{x}_1(t), \tag{21b}$$

$$\boldsymbol{x}_3(t+1) = \boldsymbol{x}_2(t), \tag{21c}$$

where $\boldsymbol{M}_\wedge$, $\boldsymbol{M}_\oplus$, $\boldsymbol{M}_\neg$ are the structure matrices of the *AND*, *XOR* and *NOT* operators, respectively.

We want to use (21) to deduce an equation of the form

$$\boldsymbol{X}(t+1) = \boldsymbol{T}\,\boldsymbol{X}(t), \tag{22}$$

where

$$\boldsymbol{X}(t+1) = \boldsymbol{x}_1(t+1) \ltimes \boldsymbol{x}_2(t+1) \ltimes \boldsymbol{x}_3(t+1), \tag{23}$$

$$\boldsymbol{X}(t) = \boldsymbol{x}_1(t) \ltimes \boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t). \tag{24}$$

Are the exact state vectors [3, 6] taking the form (17). The matrix $\boldsymbol{T}$ is the *transition matrix* of the Boolean network [3, 6]. This matrix is named **L** and called the *structure matrix* by the STP community [11]. It can be used to construct the *full state transition diagram* of the network, and to make subtle predications of both the *transient* behavior and *cyclic* behavior of the network.

We now reproduce a derivation of $\boldsymbol{X}(t+1)$ in terms of $\boldsymbol{X}(t)$ from [11, p. 118], but with the several not-so-obvious gaps therein being filled with clarifying details.

$$\begin{aligned}\boldsymbol{X}(t+1) &= \boldsymbol{x}_1(t+1) \ltimes \boldsymbol{x}_2(t+1) \ltimes \boldsymbol{x}_3(t+1)\\ &= \left(\boldsymbol{M}_{\wedge} \ltimes \boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t)\right) \ltimes \left(\boldsymbol{M}_{\neg} \ltimes \boldsymbol{x}_1(t)\right) \ltimes \boldsymbol{x}_2(t)\\ &= \boldsymbol{M}_{\wedge} \ltimes \left(\left(\boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t)\right) \ltimes \boldsymbol{M}_{\neg}\right) \ltimes \boldsymbol{x}_1(t) \ltimes \boldsymbol{x}_2(t)\\ &= \boldsymbol{M}_{\wedge} \ltimes ((\boldsymbol{I}_4 \otimes \boldsymbol{M}_{\neg}) \ltimes \left(\boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t)\right) \ltimes \boldsymbol{x}_1(t) \ltimes \boldsymbol{x}_2(t).\end{aligned}$$

Now

$$\begin{aligned}&\left(\boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t)\right) \ltimes \boldsymbol{x}_1(t) \ltimes \boldsymbol{x}_2(t)\\ &= \boldsymbol{W}_{[2,4]} \ltimes \boldsymbol{x}_1(t) \ltimes \left(\boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t)\right) \ltimes \boldsymbol{x}_2(t)\\ &= \boldsymbol{W}_{[2,4]} \ltimes \boldsymbol{x}_1(t) \ltimes \boldsymbol{x}_2(t) \ltimes \left(\boldsymbol{x}_3(t) \ltimes \boldsymbol{x}_2(t)\right)\\ &= \boldsymbol{W}_{[2,4]} \ltimes \boldsymbol{x}_1(t) \ltimes \boldsymbol{x}_2(t) \ltimes \boldsymbol{W}_{[2,2]} \ltimes \boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t)\\ &= \boldsymbol{W}_{[2,4]} \ltimes (\boldsymbol{I}_4 \otimes \boldsymbol{W}_{[2,2]}) \ltimes \left(\boldsymbol{x}_1(t) \ltimes \boldsymbol{x}_2(t)\right) \ltimes \left(\boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t)\right)\\ &= \boldsymbol{W}_{[2,4]} \ltimes \left(\boldsymbol{I}_4 \otimes \boldsymbol{W}_{[2,2]}\right) \ltimes \boldsymbol{x}_1 \ltimes \left(\boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_2(t)\right) \ltimes \boldsymbol{x}_3(t)\\ &= \boldsymbol{W}_{[2,4]} \ltimes (\boldsymbol{I}_4 \otimes \boldsymbol{W}_{[2,2]}) \ltimes \boldsymbol{x}_1(t) \ltimes \left(\boldsymbol{M}_r \ltimes \boldsymbol{x}_2(t)\right) \ltimes \boldsymbol{x}_3(t)\\ &= \boldsymbol{W}_{[2,4]} \ltimes \left(\boldsymbol{I}_4 \otimes \boldsymbol{W}_{[2,2]}\right) \ltimes (\boldsymbol{I}_2 \otimes \boldsymbol{M}_r) \ltimes \boldsymbol{x}_1(t) \ltimes \boldsymbol{x}_2(t) \ltimes \boldsymbol{x}_3(t)\\ &= \boldsymbol{W}_{[2,4]} \ltimes \left(\boldsymbol{I}_4 \otimes \boldsymbol{W}_{[2,2]}\right) \ltimes (\boldsymbol{I}_2 \otimes \boldsymbol{M}_r) \ltimes \boldsymbol{X}(t)\end{aligned}$$

$$\begin{aligned}\boldsymbol{X}(t+1) &= \boldsymbol{M}_{\wedge} \ltimes (\boldsymbol{I}_4 \otimes \boldsymbol{M}_{\neg}) \ltimes \boldsymbol{W}_{[2,4]} \ltimes \left(\boldsymbol{I}_4 \otimes \boldsymbol{W}_{[2,2]}\right) \ltimes (\boldsymbol{I}_2 \otimes \boldsymbol{M}_r) \ltimes \boldsymbol{X}(t) \quad \text{(22a)}\\ &= \boldsymbol{T} \ltimes \boldsymbol{X}(t) = \boldsymbol{T}\,\boldsymbol{X}(t).\end{aligned}$$

The transition matrix of the network is given by

$$\boldsymbol{T} = \boldsymbol{M}_{\wedge_{2\times 4}} \ltimes (\boldsymbol{I}_4 \otimes \boldsymbol{M}_{\neg})_{8\times 8} \ltimes \boldsymbol{W}_{[2,4]_{8\times 8}} \ltimes \left(\boldsymbol{I}_4 \otimes \left[\boldsymbol{W}_{[2,2]}\right]\right)_{16\times 16} \ltimes (\boldsymbol{I}_2 \otimes \boldsymbol{M}_r)_{8\times 4}, \quad (25)$$

where

Table (1). Structure matrices of seven binary Boolean operators for conventional truth-table order.

Operator	Symbol (O)	Abbreviated structure matrix	Expanded structure matrix for the basis: $[\overline{x}_1\overline{x}_2 \;\; \overline{x}_1 x_2 \;\; x_1\overline{x}_2 \;\; x_1 x_2]^T$	$x_1 \mathrm{O}\, x_2$
AND	$\wedge$	$\boldsymbol{\delta}_2[\mathbf{1},\mathbf{1},\mathbf{1},\mathbf{2}]$	$\begin{bmatrix}\mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}\end{bmatrix}$	$\begin{bmatrix}\overline{x}_1\overline{x}_2+\bar{x}_1x_2+x_1\bar{x}_2\\ x_1x_2\end{bmatrix}$
OR	$\vee$	$\boldsymbol{\delta}_2[\mathbf{1},\mathbf{2},\mathbf{2},\mathbf{2}]$	$\begin{bmatrix}\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1}\end{bmatrix}$	$\begin{bmatrix}\bar{x}_1\bar{x}_2\\ \overline{x}_1x_2+x_1\bar{x}_2+x_1x_2\end{bmatrix}$
NAND	$\uparrow$	$\boldsymbol{\delta}_2[\mathbf{2},\mathbf{2},\mathbf{2},\mathbf{1}]$	$\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{1}\\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{0}\end{bmatrix}$	$\begin{bmatrix}x_1x_2\\ \overline{x}_1\overline{x}_2+\bar{x}_1x_2+x_1\bar{x}_2\end{bmatrix}$
NOR	$\downarrow$	$\boldsymbol{\delta}_2[\mathbf{2},\mathbf{1},\mathbf{1},\mathbf{1}]$	$\begin{bmatrix}\mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{1}\\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{0}\end{bmatrix}$	$\begin{bmatrix}\overline{x}_1x_2+x_1\bar{x}_2+x_1x_2\\ \bar{x}_1\bar{x}_2\end{bmatrix}$
XOR	$\oplus$	$\boldsymbol{\delta}_2[\mathbf{1},\mathbf{2},\mathbf{2},\mathbf{1}]$	$\begin{bmatrix}\mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1}\\ \mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0}\end{bmatrix}$	$\begin{bmatrix}\overline{x}_1\overline{x}_2+x_1x_2\\ \overline{x}_1x_2+x_1\overline{x}_2\end{bmatrix}$
XNOR **Equivalence** **Bi-implication**	$\odot$ $\equiv$ $\leftrightarrow$	$\boldsymbol{\delta}_2\ [\mathbf{2},\mathbf{1},\mathbf{1},\mathbf{2}]$	$\begin{bmatrix}\mathbf{0} & \mathbf{1} & \mathbf{1} & \mathbf{0}\\ \mathbf{1} & \mathbf{0} & \mathbf{0} & \mathbf{1}\end{bmatrix}$	$\begin{bmatrix}\overline{x}_1x_2+x_1\overline{x}_2\\ \overline{x}_1\overline{x}_2+x_1x_2\end{bmatrix}$
Implication	$\rightarrow$	$\boldsymbol{\delta}_2[\mathbf{2},\mathbf{2},\mathbf{1},\mathbf{2}]$	$\begin{bmatrix}\mathbf{0} & \mathbf{0} & \mathbf{1} & \mathbf{0}\\ \mathbf{1} & \mathbf{1} & \mathbf{0} & \mathbf{1}\end{bmatrix}$	$\begin{bmatrix}x_1\overline{x}_2\\ \overline{x}_1\overline{x}_2+\overline{x}_1x_2+x_1x_2\end{bmatrix}$

$$\boldsymbol{C}_{8\times 4} = (\boldsymbol{I}_2 \otimes \boldsymbol{M}_{\mathrm{r}}) = \left(\begin{bmatrix}\mathbf{1} & 0\\ 0 & \mathbf{1}\end{bmatrix} \otimes \begin{bmatrix}\mathbf{1} & 0\\ 0 & 0\\ 0 & 0\\ 0 & \mathbf{1}\end{bmatrix} \right) = \begin{bmatrix} \mathbf{1} & 0 & \vdots & 0 & 0\\ 0 & 0 & \vdots & 0 & 0\\ 0 & 0 & \vdots & 0 & 0\\ 0 & \mathbf{1} & \vdots & 0 & 0\\ \cdots & \cdots & \vdots & \cdots & \cdots\\ 0 & 0 & \vdots & \mathbf{1} & 0\\ 0 & 0 & \vdots & 0 & 0\\ 0 & 0 & \vdots & 0 & 0\\ 0 & 0 & \vdots & 0 & \mathbf{1} \end{bmatrix}_{8\times 4}$$

$$\boldsymbol{D}_{16\times16} = \left(\boldsymbol{I}_4 \otimes \left[\boldsymbol{W}_{[2,2]}\right]\right) = \left(\begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{bmatrix} \otimes \begin{bmatrix} \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} \end{bmatrix}\right)$$

$$= \begin{bmatrix}
\mathbf{1} & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & 0 & \mathbf{1} & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & \mathbf{1} & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \mathbf{1} & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \vdots & \mathbf{1} & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & \mathbf{1} & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & \mathbf{1} & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & \mathbf{1} & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & \mathbf{1} & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & \mathbf{1} & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & \mathbf{1} & 0 & 0 & \vdots & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & \mathbf{1} & \vdots & 0 & 0 & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & \mathbf{1} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & \mathbf{1} & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & 0 & \vdots & 0 & 0 & 0 & \mathbf{1} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}_{16\times16}$$

$$\boldsymbol{E}_{16\times 8} = \boldsymbol{D}_{16\times 16} \ltimes \boldsymbol{C}_{8\times 4} = \boldsymbol{D}_{16\times 16}(\boldsymbol{C}\otimes\boldsymbol{I}_2) = \boldsymbol{D}_{16\times 16}\begin{bmatrix}
\mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1}\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}$$

$$= \begin{bmatrix}
\mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0\\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1}\\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}_{16\times 8}$$

$$\boldsymbol{F}_{16\times 8} = \boldsymbol{W}_{[2,4]_{8\times 8}} \ltimes \boldsymbol{E}_{16\times 8} = \left(\boldsymbol{W}_{[2,4]}\otimes\boldsymbol{I}_2\right)_{16\times 16}\boldsymbol{E}_{16\times 8} \qquad (26)$$

where $(\boldsymbol{W}_{[2,4]}\otimes\boldsymbol{I}_2)$ is given by

$$
\begin{bmatrix}
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
\mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}
$$

$$
\boldsymbol{F}_{16\times 8} = \begin{bmatrix}
\mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots \\
0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 \\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 \\
\cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots & \vdots & \cdots & \cdots \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\
0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{bmatrix}_{16\times 8}
$$

$$(\boldsymbol{I}_4 \otimes \boldsymbol{M}_{\neg}) = \begin{bmatrix} 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 1 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 1 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}_{8\times 8}$$

$$\boldsymbol{G}_{16\times 8} = (\boldsymbol{I}_4 \otimes \boldsymbol{M}_{\neg})_{8\times 8} \ltimes \boldsymbol{F}_{16\times 8} = \left((\boldsymbol{I}_4 \otimes \boldsymbol{M}_{\neg}) \otimes \boldsymbol{I}_2\right)_{16\times 16} \boldsymbol{F}_{16\times 8} \tag{27}$$

$where$ $(\boldsymbol{I}_4 \otimes \boldsymbol{M}_{\neg}) \otimes \boldsymbol{I}_2$ is given by

$$\begin{bmatrix} \cdots & \cdots & \cdots & & & & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 1 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 1 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 1 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 1 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}_{16\times 16}$$

$$\boldsymbol{G}_{16\times 8}=\left[\begin{array}{ccccccccccccccc}
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&\mathbf{1}&\vdots&0&\vdots&0&\vdots&0\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
\mathbf{1}&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&\mathbf{1}&\vdots&0&\vdots&0\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
0&\vdots&\mathbf{1}&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&\mathbf{1}&\vdots&0\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
0&\vdots&0&\vdots&\mathbf{1}&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&\mathbf{1}\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
0&\vdots&0&\vdots&0&\vdots&\mathbf{1}&\vdots&0&\vdots&0&\vdots&0&\vdots&0\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots
\end{array}\right]_{16\times 8}$$

$$\boldsymbol{T}_{8\times 8}=(\boldsymbol{M}_{\wedge})_{2\times 4}\ltimes\ G_{16\times 8}=((\boldsymbol{M}_{\wedge})_{2\times 4}\otimes\boldsymbol{I}_4)_{8\times 16}\boldsymbol{G}_{16\times 8}$$

where $(\boldsymbol{M}_{\wedge})_{2\times 4}\otimes\boldsymbol{I}_4$ is given by

$$\left[\begin{array}{ccccccccccccccccccc}
\mathbf{1}&0&0&0&\vdots&\mathbf{1}&0&0&0&\vdots&\mathbf{1}&0&0&0&\vdots&0&0&0&0\\
0&\mathbf{1}&0&0&\vdots&0&\mathbf{1}&0&0&\vdots&0&\mathbf{1}&0&0&\vdots&0&0&0&0\\
0&0&\mathbf{1}&0&\vdots&0&0&\mathbf{1}&0&\vdots&0&0&\mathbf{1}&0&\vdots&0&0&0&0\\
0&0&0&\mathbf{1}&\vdots&0&0&0&\mathbf{1}&\vdots&0&0&0&\mathbf{1}&\vdots&0&0&0&0\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\
0&0&0&0&\vdots&0&0&0&0&\vdots&0&0&0&0&\vdots&\mathbf{1}&0&0&0\\
0&0&0&0&\vdots&0&0&0&0&\vdots&0&0&0&0&\vdots&0&\mathbf{1}&0&0\\
0&0&0&0&\vdots&0&0&0&0&\vdots&0&0&0&0&\vdots&0&0&\mathbf{1}&0\\
0&0&0&0&\vdots&0&0&0&0&\vdots&0&0&0&0&\vdots&0&0&0&\mathbf{1}\\
\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots
\end{array}\right]_{8\times 16}$$

$$\boldsymbol{T}=\begin{bmatrix} 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & \mathbf{1} & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & \mathbf{1} & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \mathbf{1} & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & \mathbf{1} & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & \mathbf{1} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 & \vdots & 0 & 0 \\ 0 & 0 & \vdots & 0 & \mathbf{1} & \vdots & 0 & 0 & \vdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \end{bmatrix}$$

Equation (26) expresses the transition matrix T when both its rows and columns are referenced in the basis vector (17) of conventional order, which corresponds to a state vector of the form

$$[000 \quad 001 \quad 010 \quad 011 \quad 100 \quad 101 \quad 110 \quad 111]^T.$$

The matrix $\boldsymbol{T}$ can be used to construct the network state diagram shown in Fig. (1). Our matrix $\boldsymbol{T}$ is equivalent to the structure matrix L in [11, p. 118]. However, L has both its rows and columns referenced in a basis vector that is equivalent to the one in (17) but in reverse order.

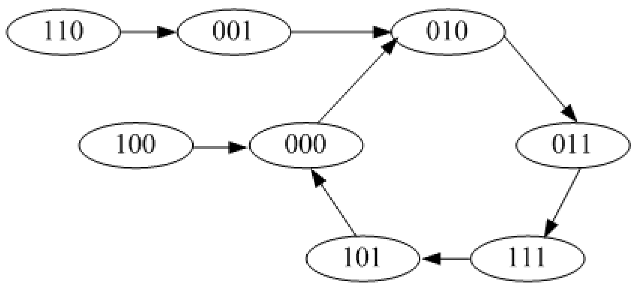

Fig. (1). The state diagram or map of all possible trajectories of the states $x_1(t)x_2(t)x_3(t)$.

4. Conclusions

Despite the extensive computational successes of the STP approach, it is still very slowly being accepted and assimilated by the scientific community. In particular, the STP concepts, techniques, and applications have not found their way yet to popular textbooks, though they are definitely expected and needed to do so sooner or later.

This paper is an attempt to popularize the STP approach by offering a detailed lengthy example of one of its prominent and most successful applications, namely, that of the analysis of synchronous Boolean networks. Besides overcoming the barrier against learning and utilizing STP concepts, the current exposition paves the way towards an efficient implementation for the STP methodology when handling binary matrices. Though the STP methodology is originally intended to handle data over the real and complex fields, the current exposition clearly indicates that the STP methodology can be directly tailored to efficiently handle the binary OR and XOR operations in Boolean algebras. Such operations intrinsically demand the use of single logic gates and should not be implemented as real addition demanding the use of an array of full adders.

5. References

[1] **Heidel, J., Maloney, J., Farrow, J. and Rogers, J.**, "Finding cycles in synchronous Boolean networks with applications to biochemical systems," *International Journal Bifurcation and Chaos*, Vol. 13, No. 3, 2003, pp. 535-552.

[2] **Farrow, C., Heidel, J., Maloney, H., and Rogers, J.**, "Scalar equations for synchronous Boolean networks with biological applications," *IEEE Transactions on Neural Networks*, Vol. 15, No. 2, 2004, pp. 348–354.

[3] **Rushdi, A. M., and Al-Otaibi, S. O.**, "On the linear analysis of synchronous switching networks," *Journal of King Abdulaziz University: Engineering Sciences*, Vol. 18, No. 2, 2007, pp. 43-72.

[4] **Rushdi, A. M., and Al-Otaibi, S. O.**, "On limitations of using scalar equations for analyzing synchronous Boolean networks," *Journal of King Abdulaziz University: Engineering Sciences*, Vol. 19, No. 2, 2008, pp. 41-49.

[5] **Rushdi, A. M., and Alsogati, A. A.,** "On reduced scalar equations for synchronous Boolean networks," *Journal of Mathematics and Statistics*, Vol. 9, No. 3, 2013, pp. 262-276.

[6] **Rushdi, A. M., and Alsogati, A. A.,** "Matrix analysis of synchronous Boolean networks," *International Magazine on Advances in Computer Science and Telecommunications* (*IMACST*), Vol. 6, 2015.

[7] **Zheng, D., Yang, G., Li, X., and Wang, Z.**, "An algorithm to find cycles of biochemical systems," *International Conference on Information, Business and Education Technology (ICIBET), Atlantis Press*, 2013, pp. 14–18.

[8] **Cheng, D.,** "Input-state approach to Boolean networks," *IEEE Transactions on Neural Networks*, Vol. 20, No. 3, 2009, pp. 512–521.

[9] **Cheng, D., and Qi, H.**, "State–space analysis of Boolean networks," *IEEE Transactions on Neural Networks*, Vol. 21, No. 4, 2010, pp. 584-594.

[10] **Cheng, D., and Qi, H.**, "A linear representation of dynamics of Boolean networks," *IEEE Transactions on Automatic Control*, Vol. 55, No. 10, 2010, pp. 2251–2258.

[11] **Cheng, D., Qi, H., and Z. Q. Li**, *Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach*, Springer-Verlag, London, 2011.

[12] **Cheng, D., Qi, H., and Zhao, Y.**, "On Boolean control networks-An algebraic approach," *Proceedings of the 18th IFAC "International Federation of Automatic Control" World Congress Milano (Italy)*, 2011, pp. 8366-8377.

[13] **Laschov, D., and Margaliot, M.**, "A maximum principle for single-input Boolean control networks," *IEEE Transactions on Automatic Control*, Vol. 56, No. 4, 2011, pp. 913-917.

[14] **Cheng, D., Qi, H., and Zhiqiang, L.**, "Model construction of Boolean networks via observed data," *IEEE Transactions on Neural Networks*, Vol. 22, No. 4, 2011, pp. 525-536.

[15] **Li, Z., Song, J., and Xiao, H.**, On the cycles of Boolean networks. *IEEE 24th Chinese Control and Decision Conference (CCDC)*, 2012, pp. 770-774.

[16] **Li, R., and Chu, T.**, "Complete synchronization of Boolean networks," *IEEE Transactions on Neural Networks Learning Systems*," Vol. 23, No. 5, 2012, pp. 840-846.

[17] **Zhan, J., Lu, S., and Yang, G.**, "Analysis of Boolean networks using an optimized algorithm of structure matrix based on semi-tensor product," *Journal of Computers,* Vol. 8, No. 6, 2013, pp. 1441-1448.

[18] **Li, R., Yang, M., and Chu, T.**, "Synchronization design of Boolean networks via the semi-tensor product method," *IEEE Transactions on Neural Networks Learning Systems,* Vol. 24, No. 6, 2013, pp. 996-1001.

[19] **Cheng, D., Qi, H., and Xue**, **A.,** "A survey of semi-tensor product of matrices," *Journal of System Science and Complexity*, Vol. 20, 2007, pp. 304-322.

[20] **Cheng, D., and Zhang L., "**On semi-tensor product of matrices and its applications," *Acta Mathematicae Applicatae Sinica (English Series)*, Vol. 19, No. 2, 2003, pp. 219-228.

[21] **Cheng, D.,** "Semi-tensor product of matrices and its applications: A survey," *Proceeding of the 6th ICCM' 07, Zhejiang, China,* Vol. 3, 2007, pp. 641-668.

[22] **Rushdi, A. M., and Ghaleb, F. A.**, "A tutorial exposition of semi-tensor products of matrices with a stress on their representation of Boolean functions," *Journal of King Abdulaziz University: FCIT*, Vol. 2, No. 1, 2013, pp. 1-39.

[23] **Brewer, J.,** "Kronecker products and matrix calculus in system theory," *IEEE Transactions on Circuits and Systems*, Vol. CAS-25, No. 9, 1978, pp. 772-781, corrections: *ibid*, Vol. CAS-26, No. 5, 1979, p. 360.

[24] **Graham, A.,** *Kronecker Products and Matrix Calculus: with Applications*, Ellis Horwood Limited and John Wiley & Sons, New York, NY, USA, 1981.

[25] **Rushdi, A. M., and F. A. M. Ghaleb**, "On self-inverse binary matrices over the binary Galois field," *Journal of Mathematics and Statistics*, Vol. 9, No. 3, 2013, pp. 238-248.

[26] **Zhoa, Y., Gao, X., and Cheng, D.,** "Semi-tensor product approach to Boolean functions," pp. *1-9,* Unpublished Preprint, Available at http://lsc.amss.ac.cn/~dcheng/preprint/bf01.pdf

[27] **Cheng, D., and Qi, H.,** "Matrix expression of logic and fuzzy control," *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference, (CDC-ECC)*, 2005, pp. 3273-3278.

[28] **Cheng, D., and Li, Z.,** "Solving logic equation via matrix expression," *Frontiers of Electrical and Electronic Engineering in China*, Vol. 4**,** No**.** 3, 2009, pp. 259-269.

[29] **Cheng, D., Yin, Z., and Xiangru, X.,** "Matrix approach to Boolean calculus," *Proceedings of the 50th IEEE Conference on Decision and Control and the European Control Conference (CDC-ECC)*, 2011, pp. 6950-6955.

[30] **Cheng, D., and Qi, H.,** "Logic and logic-based control," *Journal of Control Theory and Applications*, Vol. 6, No. 1, 2012, pp. 26-36.

[31] **Rushdi, A. M.,** "Development of modified nodal analysis into a pedagogical tool," *IEEE Transactions on Education*, Vol. E-28, No. 1, 1985, pp. 17-25.

[32] **Muroga, S.,** *Logic Design and Switching Theory*, Wiley, New York, NY, USA, 1979.

شرح تعليمي لتحليل الشبكات البولانية المتزامنة باستخدام المضروبات شبه المنظومية

علي محمد علي رشدي و فارس أحمد محمد غالب

قسم الهندسة الكهربائية وهندسة الحاسبات، كلية الهندسة،

ص. ب. ٨٠٢٠٤، جامعة الملك عبد العزيز، جدة، ٢١٥٨٩، المملكة العربية السعودية

arushdi@kau.edu.sa

(قدم للنشر في ٢٠١٤/٩/١٣ ، قبل للنشر في ٢٠١٥/٣/٢٥)

ملخص البحث. تمثل ورقة البحث هذه شرحا تعليميا مفصلا لتحليل الشبكات البولانية المتزامنة باستخدام نوع جديد من مضروبات المصفوفات يسمى المضروب شبه المنظومي (ض ش ن). يقوم هذا المضروب بإيجاد حاصل لضرب مصفوفتين أم ن و بف ك فيهما عدد الأعمدة ن للمصفوفة الأولى ليس بالضرورة مساويا لعدد الصفوف ف للمصفوفة الثانية، وإنما قد يكون مضاعفا له أو قد يكون قاسما له. إن فضاء الحالات لشبكة بولانية مؤلفة من n من الرؤوس يمثل هنا بمتجه يحوي عدد 2^n من الحالات مرتبة ترتيبا طبيعيا يتم إيجادها كمضروب شبه منظومي لعدد n من المتجهات ذوات العنصرين تمثل متغيرات الشبكة. إن أحد الإسهامات الملحوظة لورقة البحث هذه هو تعبيرها المصفوفي عن المنطق بالترتيب الاصطلاحي المألوف في جداول الصدق وليس بالترتيب المعكوس المستغرب الذي دأب الباحثون في المضروبات شبه المنظومية على استخدامه. نقوم هنا بإعادة إنتاج التحليل بالمضروبات شبه المنظومية لشبكة تعد مثالا نمطيا تقليديا. يشتمل تحليلنا على تفصيلات دقيقة تجعل مسألة المعالجات بالمضروبات شبه المنظومية أيسر منالا وأقرب فهما لمستخدميها المحتملين. يشير تحليلنا إلى طرائق أكثر كفاية وأفضل سرعة لتنفيذ الحل بالمضروبات شبه المنظومية لا نغرم فيه كلفة لمؤثر "أو المشتملة" أو لمؤثر "أو المستثنية" أكثر مما يستحقان في واقع الأمر.