International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

Matrix Analysis of Synchronous Boolean Networks

Ali Muhammad Ali Rushdi
Department of Electrical and Computer Engineering,
King Abdulaziz University,
P. O. Box 80200, Jeddah 21589, Saudi Arabia.
Corresponding author: arushdi@kau.edu.sa

Adnan Ahmad Alsogati
Department of Electrical and Computer Engineering,
King Abdulaziz University,
P. O. Box 80200, Jeddah 21589, Saudi Arabia.
E-mail: alsogati@hotmail.com

(Received July 11, 2020; Accepted October 5, 2020)

Abstract
The synchronous Boolean network (SBN) is a simple and powerful model for describing, analyzing, and simulating cellular biological networks. This paper seeks a complete understanding of the dynamics of such a model by employing a matrix method that relies on relating the network transition matrix to its function matrix via a self-inverse state matrix. A recursive ordering of the underlying basis vector leads to a simple recursive expression of this state matrix. Hence, the transition matrix is computed via multiplication of binary matrices over the simplest finite (Galois) field, namely the binary field GF(2), i.e., conventional matrix multiplication involving modulo-2 addition, or XOR addition. We demonstrate the conceptual simplicity and practical utility of our approach via an illustrative example, in which the transition matrix is readily obtained, and subsequently utilized (via its powers, characteristic equation, minimal equation, 1-eigenvectors, and 0-eigenvectors) to correctly predict both the transient behavior and the cyclic behavior of the network. Our matrix approach for computing the transition matrix is superior to the approach of scalar equations, which demands cumbersome manipulations and might fail to predict the exact network behavior. Our approach produces result that exactly replicate those obtained by methods employing the semi-tensor product (STP) of matrices, but achieves that without sophisticated ambiguity or unwarranted redundancy.

Key words- Synchronous Boolean networks, Transition matrix, Function matrix, Recursive ordering, Self-inverse state matrix, Galois field GF (2), Characteristic equation, Minimal equation, 1-eigenvectors, 0-eigenvectors.

1. Introduction

Many biological systems, such as cellular networks and genetic regulatory networks, are too complex to allow exact or nearly exact analysis. Therefore, these systems are usually studied in terms of synchronous Boolean networks (SBNs) as approximating models. In fact, an SBN is the simplest possible conceptual model that mimics or captures the essential features of the original systems, and that can be effectively used for describing, analyzing, and simulating these systems. An SBN is 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. Since the total number of possible network states is finite (2^n) and the network changes states sequentially in discrete time steps, the network must necessarily return to a previous state in a finite time of at most 2^n time points. All possible trajectories of the network are either cycles (loops or attractors) of any length from size 1 (a fixed point) to a maximum length of 2^n, or transient states leading eventually to a cycle (attractor).

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

A total description of an SBN is typically achieved by solving a matrix equation, or by algorithms in which a matrix equation is implicit. A simpler description is possible when such a matrix equation is replaced by a scalar equation or a reduced scalar equation (Rushdi and Al-Otaibi, 2007; Rushdi, 2015). Unfortunately, the scalar-equation technique suffers from several shortcomings and limitations (Rushdi, 2015). Recently, the dynamics of SBNs have been explored by a novel matrix method utilizing a new matrix product, called the semi-tensor product (STP) of matrices (Cheng, et al., 2011; Li and Chu, 2012; Rushdi and Ghaleb, 2016; Cheng, 2019). The STP approach uses a certain matrix expression of logic for the numerical derivation of the transition matrix $[T]$ of the Boolean network. This matrix is then analyzed to deduce full information about the transient and cyclic behavior of the network. It is noted that the concepts of STP are quite involved, and though they lead to useful computational algorithms, they are not easy to grasp, and suffer from unwarranted duplications.

The purpose of this paper is to derive the transition matrix $[T]$, independently from any STP formulations or derivations, through the multiplication of binary matrices over GF(2), i.e., conventional multiplication associated with modulo-2 addition, or XOR addition. To attain this purpose, the paper utilizes well-known concepts relating the network transition matrix $[T]$ to its function matrix $[A]$ via the relation $[T] = [S_n]^{-1}[A][S_n]$, where $[S_n]$ is the state matrix of order n and dimensions $2^n \times 2^n$ (Cull, 1971; Rushdi and Al-Otaibi, 2007). The function matrix $[A]$ is readily obtained from knowledge of the next-state scalar functions and their various products. If the underlying basis vector is rendered a recursive rather than a lexicographic ordering, a neat recursive definition of the state matrix $[S_n]$ results. Since $[S_n]$ can be constructed easily (via this recursive definition) and is inverted at no cost (being a self-inverse matrix satisfying $[S_n]^{-1} = [S_n]$), the computation of $[T]$ in terms of $[S_n]$ and $[A]$ is straightforward. Once $[T]$ is obtained, the complete state diagram can be drawn. However, subtle inferences about the network behavior are possible without explicitly drawing the state diagram by exploring some properties of $[T]$ such as its powers, characteristic equation, minimal equation, 1-eigenvectors and 0-eigenvectors.

The organization of the remainder of this paper is as follows. Section 2 presents the methodology adopted herein. It starts by reviewing pertinent quantities and relations. Then, it presents our algorithm for computing the transition matrix $[T]$. Finally, it explains how $[T]$ can be used to deduce the cyclic and transient behavior of the network without resorting to a construction of the network full state diagram. Section 3 presents our results in the form of an illustrative example of a synchronous Boolean network previously studied in the literature. The example not only demonstrates the binary matrix multiplications over GF(2) needed in constructing $[T]$ and its powers, but it also illustrates mathematical techniques employed in deriving necessary related entities. Section 4 discusses our findings via this example and asserts that our algorithm replicates in clear and succinct steps the results obtained by the STP methodology. Our transition-matrix predications of network transient and cyclic behavior are in full agreement with those obtained via a construction of the network state diagram. Section 5 concludes the paper.

2. Methodology

The methodology adopted herein is one of rigorous mathematical analysis and further demonstration by way of examples. In subsection 2.1, we present preliminary definitions and, in particular, identify four different forms of the basis vectors that are used in the representation of two-valued Boolean functions. These are the lexicographic basis (Cull, 1971), the recursive basis (Rushdi and Al-Otaibi, 2007), the conventional truth-table basis (Rushdi, 2018; Rushdi and Rushdi, 2018), and the STP basis (Cheng, 2019; Rushdi and Ghaleb, 2016). Subsection 2.2

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

outlines our algorithm for constructing the transition matrix $[T]$, while subsection 2.3 reviews how $[T]$ could be utilized in predicting network behavior via its powers, characteristic equation, minimal equation, 1-eigenvectors and 0-eigenvectors.

2.1 Preliminary Definitions

Functions used herein are defined over the simplest finite (Galois) field GF (2). The field has only two elements: the additive identity (0) and the multiplicative identity (1). The field addition $(+)$ and multiplication $(\times)$ are defined as a modulo-2 addition and a 1-bit multiplication, which resemble the *exclusive-OR* $(\oplus)$ and the ANDing $(\wedge)$ operations, respectively, in two-valued Boolean algebra. Any function of n variables over GF(2) has a linear or Boolean-ring representation via a polynomial of 2^n terms (called a Taylor, a Zhegalkin or a Reed-Müller polynomial), or via a vector of length 2^n comprising binary coefficients of this polynomial. In the sequel, we interchangeably use two isomorphic representations: the one over GF(2) and that of the Boolean ring, and in particular, our $(+)$ sign means modulo-2 addition or *XOR* operation, and our $(\times)$ sign (to be omitted and indicated simply by juxtaposition) depicts 1-bit multiplication or *AND* operation. A vector of 2^n terms constitutes a basis for all functions of n variables over GF(2) or equivalently all switching functions of n variables, in the sense that any of these functions equals the (scalar) dot product of the vector of binary coefficients representing it with the afore-mentioned basis vector. Cull (1971) proposed the use of a *lexicographic* order for the basis vector:

$$\vec{C}_n = [1 \quad x_1 \quad x_2 \quad x_3 \quad \dots \quad x_n \quad x_1x_2 \quad x_1x_3 \quad \dots \quad x_1x_n \quad x_2x_3 \quad \dots \quad x_{n-1}x_n \quad x_1x_2x_3 \quad \dots \quad x_{n-2}x_{n-1}x_n \quad \dots \quad x_1x_2x_3 \dots x_n]^T. \tag{1}$$

In fact, Cull replaced the leading 1 in (1) with a 0; an oversight since 0 is not a logical product at all, while 1 is the multiplicative identity (product of 0 variables). Rushdi and Al-Otaibi (2007) proposed a recursive order:

$$\vec{B}_n = [1 \quad x_1 \quad x_2 \quad x_1x_2 \quad x_3 \quad x_1x_3 \quad x_2x_3 \quad x_1x_2x_3 \quad .. \quad x_n \quad x_1x_n \quad x_2x_n \quad x_1x_2x_n \quad .. \quad x_1x_2x_3 \dots x_n]^T, \tag{2}$$

which can be defined by the following recursive relation and boundary condition:

$$\vec{B}_n = \begin{bmatrix} \vec{B}_{n-1} \\ \vec{B}_{n-1} \wedge x_n \end{bmatrix}, \qquad \vec{B}_0 = [1]. \tag{3}$$

An implication of the recursive relation in (3) is that none of the logical products (that are elements of $\vec{B}_n$) is succeeded by products that are subsumed by it. We recall that a product subsumes another if the set of literals of the former is a superset of that of the latter (Rushdi and Rushdi, 2017). In the sequel, we will always implicitly use the recursive ordering in (3). We stress the existence of other ways of defining the basis vector that are in common use. A prominent one is the common truth-table or minterm order that can be given recursively as

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

$$\vec{M}_n = \vec{M}_{n-1} \otimes \begin{bmatrix} \bar{x}_n \\ x_n \end{bmatrix}, \qquad \vec{M}_0 = [1], \tag{4}$$

where $\otimes$ is the Kronecker product for two matrices. For $n = 3$, this basis vector is:

$$\vec{M}_3 = [\bar{x}_1\bar{x}_2\bar{x}_3 \quad \bar{x}_1\bar{x}_2x_3 \quad \bar{x}_1x_2\bar{x}_3 \quad \bar{x}_1x_2x_3 \quad x_1\bar{x}_2\bar{x}_3 \quad x_1\bar{x}_2x_3 \quad x_1x_2\bar{x}_3 \quad x_1x_2x_3]^T. \tag{5}$$

Clearly, if the elements of $\vec{M}_n$ are understood to depict binary numbers for $x_1 = x_2 = x_3 = 1$, then the values of these numbers follow the natural order from 0 to $(2^n - 1)$. Another common basis used extensively in the literature is the one of expressing logical (switching) variables via semi-tensor products (STP) of matrices (Cheng, 2019; Rushdi and Ghaleb, 2016). Here, the basis is simply the STP of the pertinent variables:

$$\vec{D}_n = \ltimes_{i=1}^{n} \vec{x}_i, \tag{6}$$

where the symbol $\ltimes$ denotes the STP of matrices, and $\vec{x}_i = [x_i \quad \bar{x}_i]^T$ is a vector representing the variable x_i. This corresponds to a recursive definition and an explicit one (for $n = 3$) of the form:

$$\vec{D}_n = \vec{D}_{n-1} \otimes \begin{bmatrix} x_n \\ \bar{x}_n \end{bmatrix}, \qquad \vec{D}_0 = [1]. \tag{7}$$

$$\vec{D}_3 = [x_1x_2x_3 \quad x_1x_2\bar{x}_3 \quad x_1\bar{x}_2x_3 \quad x_1\bar{x}_2\bar{x}_3 \quad \bar{x}_1x_2x_3 \quad \bar{x}_1x_2\bar{x}_3 \quad \bar{x}_1\bar{x}_2x_3 \quad \bar{x}_1\bar{x}_2\bar{x}_3]^T. \tag{8}$$

The values of the numbers corresponding to the elements of $\vec{D}_n$ follow a *reverse* order from $(2^n - 1)$ down to 0. This reverse order results from defining $\vec{x}_i$ as $[x_i \quad \bar{x}_i]^T$ and not as $[\bar{x}_i \quad x_i]^T$. Next, we review some pertinent terminology from Rushdi and Al-Otaibi (2007), and Rushdi (2015) concerning an SBN of n nodes. Note that we use two distinct state vectors $\vec{Y}(t)$ and $\vec{X}(t)$. The STP community (see, e.g., Li et al., 2012) typically uses only one state vector that is named $\vec{X}(t)$ therein which corresponds to our $\vec{Y}(t)$.

The Exact State Vector $\vec{Y}(t)$: A binary column vector of dimension 2^n, whose elements are all 0 except one element that is of value 1, which corresponds to the position where the product depicting the network state occurs in the basis vector $\vec{B}_n$ given in (2) or (3).

The Cumulative State Vector $\vec{X}(t)$: A binary column vector of dimension 2^n, which equals the basis vector $\vec{B}_n$ evaluated at the network state at instant t. Hence, $\vec{X}(t)$ has $1's$ in the positions representing any of the products of $\vec{B}_n$ equated to 1, *i.e.*, the products subsumed by the product depicting the network state. While the number of $1's$ in the vector $\vec{Y}(t)$ is 1, this number in the vector $\vec{X}(t)$ is 2^k, where $k = 0, 1, \dots, n$ $(\vec{Y}(t) \leq \vec{X}(t))$.

The Transition Matrix $[T]$: A $2^n \times 2^n$ matrix with a single 1 and with 0's otherwise in each column (there may be more than one element of 1 in any row, and as many rows with all 0's).The transition matrix $[T]$ represents the state transition diagram of the network. The particular structure of $[T]$ asserts that the network is in only one state at a time instance and proceeds to a unique next state in the next instant. The transition matrix $[T]$ relates the exact next-state vector

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

$\vec{Y}(t+1)$ to the exact present-state one $\vec{Y}(t)$ via:

$$\vec{Y}(t+1) = [T]\,\vec{Y}(t). \tag{9}$$

If the single 1 in column j of $[T]$ is element t_{ij}, this means that state i is the next state of state j, abbreviated as:

$$\left\{\left(\vec{Y}(t) = \vec{\delta}_{2^n}^{j}\right) \Rightarrow \left(\vec{Y}(t+1) = Col_j[T] = \vec{\delta}_{2^n}^{i}\right)\right\}. \tag{10}$$

where $\vec{\delta}_{2^n}^{j}$ is a column vector of length 2^n whose elements are all $0's$ with the exception of a single 1-element in position j. The dynamics of an SBN is uniquely determined by the matrix equation (9), provided the initial state $\vec{Y}(0)$ is specified. The structure matrix $[L]$ in Cheng et al. (2011) is the same as $[T]$, but in disguise emanating from the differences in bases. The basis for $[T]$ in Cull (1971) is $\vec{C}_n$ given by (1) and the basis in Rushdi and AL-Otaibi (2007) is $\vec{B}_n$ given by (2) or (3), while it is $\vec{D}_n$ given by (6-8) for $[L]$.

The Function Matrix $[A]$: A $2^n \times 2^n$ matrix that has as its rows the vector representations of the 2^n products of the n scalar functions computed by the elements of the SBN (taken k at a time $(0 \le k \le n)$). The function matrix relates two consecutive instances of the vector $\vec{X}$ as:

$$\vec{X}(t+1) = [A]\,\vec{X}(t). \tag{11}$$

The State Matrix $[S_n]$: A $2^n \times 2^n$ upper triangular binary matrix, the columns of which are the $\vec{X}$ vectors representing all the 2^n possible states of the network. The recursive ordering of the basis $\vec{B}_n$ in (2) or (3) allows the matrix $[S_n]$ to have the recursive definition:

$$[S_n] = \begin{bmatrix} [S_{n-1}] & [S_{n-1}] \\ [0_{n-1}] & [S_{n-1}] \end{bmatrix}, \quad n > 0, \qquad [S_0] = [1]. \tag{12}$$

where $[0_{n-1}]$ is the zero matrix of dimensions $2^{n-1} \times 2^{n-1}$. This recursive definition of $[S_n]$ is reminiscent of the transformation between the linear (Reed-Müller) representation and the minterm (truth-table) representation of a switching function (Rushdi and Ghaleb, 2016). The matrix $[S_n]$ is an involutory (self-inverse) matrix, *i.e.*,

$$[S_n]^{-1} = [S_n], \tag{13}$$

$$[S_n]^2 = [S_n][S_n] = [S_n][S_n]^{-1} = [I_n], \tag{14}$$

where $[I_n]$ is the identity matrix of $2^n \times 2^n$ elements. The state matrix $[S_n]$ is used to connect the transition matrix $[T]$ and the function matrix $[A]$ via the following similarity transformations (Cull, 1971; Rushdi, 2015):

$$[A][S_n] = [S_n][T], \tag{15}$$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

$$[A] = [S_n][T][S_n]^{-1} = [S_n][T][S_n], \tag{16}$$

$$[T] = [S_n]^{-1}[A][S_n] = [S_n][A][S_n]. \tag{17}$$

The matrix $[S_n]$ can also relate the exact state vector $\vec{Y}(t)$ and the cumulative one $\vec{X}(t)$:

$$\vec{X}(t) = [S_n]\,\vec{Y}(t), \qquad \vec{Y}(t) = [S_n]\,\vec{X}(t), \tag{18}$$

The similarity in (16) and (17) between $[T]$ and $[A]$ extends to similarity between their powers $[T]^k$ and $[A]^k$, where k is any natural integer $(k = 0, 1, 2, \cdots)$, *i.e.*,

$$[T]^k = [S_n][A]^k[S_n], \quad [A]^k = [S_n][T]^k[S_n]. \tag{19}$$

2.2 Algorithm for Constructing the Transition Matrix $[T]$

- Represent the current and next state values $x_i(t)$ and $x_i(t+1)$ of the $i'th$ variable by x_i and f_i, respectively. Express next-state functions $f_i(x_1, x_2, \cdots, x_n)$, $1 \le i \le n$, in linear form, *i.e.*, in a Reed-Müller representation. Hence, construct their products, taken 0 at a time $(f_0 = 1)$, taken one at a time $(f_i, 1 \le i \le n)$, taken two at a time $(f_i f_j, 1 \le i < j \le n)$, taken three at a time $(f_i f_j f_k, 1 \le i < j < k \le n)$, and so on till the overall product $\left(\prod_{i=1}^{n} f_i\right)$ is reached. Express each product in linear form.
- Arrange the columns of the present states (x_i $products$) and the rows of the next-state functions (f_i $products$), both according to the recursive basis in (2) or (3), and construct the function matrix $[A]$ accordingly. Figure 1 illustrates such a construction for $n = 3$.
- Construct $[S_n]$ utilizing the recursive relation and boundary condition in (12).
- Use binary matrix multiplication to obtain $[T]$ via a pre-multiplication and a post-multiplication of $[A]$ by $[S_n]$, *i.e.*, via (17). As a check, the resulting $[T]$ should have a single element of value 1 in each of its binary columns. The column vectors of $[T]$ constitute a subset (with possibly repetitive elements) of the set acting as an orthonormal basis of the 2^n-dimensional vector space.

<table>
<tr><th></th><th>1</th><th>x_1</th><th>x_2</th><th>x_1x_2</th><th>x_3</th><th>x_1x_3</th><th>x_2x_3</th><th>$x_1x_2x_3$</th></tr>
<tr><td>f_0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>f_1</td><td colspan="8" rowspan="2">Next-State Functions in Linear Form</td></tr>
<tr><td>f_2</td></tr>
<tr><td>f_1f_2</td><td colspan="8">Products of Next-State Functions in Linear Form</td></tr>
<tr><td>f_3</td><td colspan="8">Next-State Functions in Linear Form</td></tr>
<tr><td>f_1f_3</td><td colspan="8" rowspan="3">Products of Next-State Functions in Linear Form</td></tr>
<tr><td>f_2f_3</td></tr>
<tr><td>$f_1f_2f_3$</td></tr>
</table>

Figure 1. Construction of the function matrix $[A]$ with recursive ordering for column and row bases.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

2.3 Cyclic and Transient behavior

We review some of the mathematics utilized in the study of the cyclic and transient behaviors of an SBN. Most of the results cited herein were reported by Cull (1971), enhanced, elaborated, and proved by Rushdi and Al- Otaibi (2007), and utilized by Li et al. (2012). A state of the network $\vec{Y}(t)$ is called cyclic if:

$$\vec{Y}(t+m) = [T]^m \vec{Y}(t) = \vec{Y}(t), \quad for\ some\ m > 0, \tag{20}$$

and, otherwise, it is called transient. A set of cyclic states that map into one another are called a cycle (attractor). An example of a cycle of length k is the set given by $C = \{\vec{Y}(t), [T]\,\vec{Y}(t), \cdots, [T]^{k-1}\,\vec{Y}(t)\}$ provided $[T]^k\,\vec{Y}(t) = \vec{Y}(t)$ and the elements of C are distinct (Cheng et al., 2011, p. 108). For $k = 1$, the set C is the singleton $\{\vec{Y}(t)\}$, and the cycle becomes of length 1, *i.e.*, a fixed point. A cycle is therefore definable as a $[T]$-invariant subspace. A state into which no state maps is called a first state, *i.e.*, a first state does not serve as a next state for any current state, and hence the row corresponding to it in the transition matrix $[T]$ is an all-0 row. A set of transient states that can be reached from a certain first state are called a transient chain. Note that a state is transient if it is a first state or it can be reached only from a first state or from some other transient states. By analogy to characteristic equations of real matrices, Cull (1971) introduced the concept of the binary characteristic equation of a binary matrix $[T]$. He defined it as $det([T] - \lambda[I]) = 0$ where the operations are carried out in the binary field where subtraction $(-)$ and addition $(+)$ are indistinguishable. In fact, subtraction is not even defined on GF(2). Hence, the characteristic equation becomes $det([T] + \lambda[I]) = 0$. This equation takes the form:

$$det([T] + \lambda[I]) = \lambda^k(\lambda^{r_1} + 1)\cdots(\lambda^{r_i} + 1) = 0, \tag{21}$$

where k is the total number of transient states and the subscripted $r's$ are the lengths of the various cycles. We note that the eigenvalues are the roots of (21), and hence the only possible eigenvalues in GF(2) are $\lambda = 1$ (corresponding to one-eigenvectors that represent cycles) and $\lambda = 0$ (corresponding to zero-eigenvectors that represent transient chains). The factorization of the binary characteristic equation is not always unique, so that one might not be able to specify all cycles exactly. Since matrices $[A]$ and $[T]$ are similar, they have the same characteristic equation:

$$det([A] + \lambda[I]) = det([T] + \lambda[I]) = 0. \tag{22}$$

The Cayley-Hamilton theorem (which states in the continuous case that any matrix satisfies its own characteristic equation) is still applicable in this finite case, *i.e.*,

$$[T]^k([T]^{r_1} + [I])\cdots([T]^{r_i} + [I]) = [A]^k([A]^{r_1} + [I])\cdots([A]^{r_i} + [I]) = [0]. \tag{23}$$

Cull (1971) asserted that there is an equation of lowest degree that $[A]$ satisfies; which he called the minimal equation of $[A]$, namely:

$$[A]^{r_0}([A]^{r_h} + [I])\cdots([A]^{r_g} + [I]) = [0]. \tag{24}$$

Here, r_0 is the length of the longest transient chain and $r_h, \cdots, r_g$ are lengths of distinct cycles such that all other cycle lengths divide one of the $r's$ with no remainder. The cyclic behavior of

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

an SBN can also be obtained from the $\vec{X}$-type 1-eigenvectors of the function matrix $[A]$ or the $\vec{Y}$-type 1-eigenvectors of the transition matrix $[T]$. In fact, the cycles of the network are linearly independent 1-eigenvectors of $[T]$, and the 1-eigenvectors of $[T]$ are cycles or the unions of cycles (Cull, 1971; Rushdi and Al-Otaibi, 2007; Li et al., 2012). Likewise, the 0-eeigenvectors can be used in the study of the transient chains. There are as many independent 0-eigenvectors as there are chains. Each of these eigenvectors is of the form $\vec{Y}_i + \vec{Y}_j$, where $\vec{Y}_i \neq \vec{Y}_j$. At least one of the states $\vec{Y}_i$ and $\vec{Y}_j$ is the last state of a transient chain, *i.e.*, a state that is not on a cycle, but whose next state is on a cycle. Cheng et al. (2011) proved that the number N_1 of fixed points for the network equals the trace (sum of diagonal elements) of $[T]$, i.e., $N_1 = tr([T])$. They also asserted that the number of states on cycles of length k which is a divisor of an integer m is:

$$tr([T]^m) = \sum_{k\ divides\ m} kN_k = mN_m + \sum_{k\ properly\ divides\ m} kN_k, \tag{25}$$

and hence the number N_m of cycles of length m $(1 \leq m \leq 2^n)$ is given inductively by:

$$N_m = \frac{1}{m}\left[tr([T]^m) - \sum_{k\ properly\ divides\ m} kN_k\right], \tag{26}$$

where the relation $N_1 = tr([T])$ is both a boundary condition and a special case for (26).The transient period r_0 of the network (the length of its longest transient chain) is the minimum power i such that $[T]^i$ appears again in the sequence $[T]^{i+1}, [T]^{i+2}, \cdots$, *i.e.*, it is minimum power in a relation of the form: $[T]^{r_0} = [T]^{r_0+u}$, where $u > 0$ is the least common multiple of the lengths of all cycles. Recently, Li et al. (2012) observed that if $rank([T] - [I]) = r$, then the network has $k = (2^n - r)$ cycles. Again, the minus operation in $([T] - [I])$ should, of course, be interpreted as modulo-2 addition, and hence one should speak about $([T] + [I])$.

3. A Demonstrative Example

We demonstrate the computational details of the algorithm in Sec. 2, and the practical utility of our formulation in predicting network behavior via an illustrative example, that was analyzed earlier via the STP method by Cheng et al. (2011) in pp. 105 and 107. We produce results that agree with those of the exhaustive state diagram, while adhering to conceptual simplicity and insightful visibility. Our example is a 3-element network whose next-states are given by: $f_1 = x_2x_3$, $f_2 = 1 + x_1$, and $f_3 = x_2 + x_3 + x_2x_3$. Now, we form the various products of these functions as: $f_1f_2 = x_2x_3 + x_1x_2x_3$, $f_1f_3 = x_2x_3$, $f_2f_3 = x_2 + x_3 + x_2x_3 + x_1x_2 + x_1x_3 + x_1x_2x_3$ and $f_1f_2f_3 = x_2x_3 + x_1x_2x_3$. Then, we construct the function matrix $[A]$ using the strategy in Figure 1.

	1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2x_3$
f_0	**1**	0	0	0	0	0	0	0
f_1	0	0	0	0	0	0	**1**	0
f_2	**1**	**1**	0	0	0	0	0	0
f_1f_2	0	0	0	0	0	0	**1**	**1**
f_3	0	0	**1**	0	**1**	0	**1**	0
f_1f_3	0	0	0	0	0	0	**1**	0
f_2f_3	0	0	**1**	**1**	**1**	**1**	**1**	**1**
$f_1f_2f_3$	0	0	0	0	0	0	**1**	**1**

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

Here, we deliberately showed the bases indexing the rows and columns of $[A]$. The state matrix $[S_3]$ is given by

	1	x_1	x_2	x_1x_2	x_3	x_1x_3	x_2x_3	$x_1x_2x_3$
1	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
x_1	0	**1**	0	**1**	0	**1**	0	**1**
x_2	0	0	**1**	**1**	0	0	**1**	**1**
x_1x_2	0	0	0	**1**	0	0	0	**1**
x_3	0	0	0	0	**1**	**1**	**1**	**1**
x_1x_3	0	0	0	0	0	**1**	0	**1**
x_2x_3	0	0	0	0	0	0	**1**	**1**
$x_1x_2x_3$	0	0	0	0	0	0	0	**1**

Now, the transition matrix is obtained via (17) as $[T] = ([S_3][A])[S_3]$, or explicitly as

$$\begin{bmatrix} 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & \mathbf{1} & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \end{bmatrix} = \begin{bmatrix} 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \end{bmatrix}. \quad (27)$$

We might abbreviate $[T]$ in (27) as $\vec{\delta}_8 = [2,\ 0,\ 6,\ 4,\ 6,\ 4,\ 7,\ 5]$. The equivalence of our $[T]$ in (27) to the structure matrix $[L]$ in p. 107 of Cheng et al. (2011) is true, albeit not obvious due to disparity in the bases used. The characteristic equation for $[T]$ is:

$$det([T] + \lambda[I]) = det\begin{bmatrix} \boldsymbol{\lambda} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \boxed{\boldsymbol{\lambda}} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & \boldsymbol{\lambda} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boxed{\boldsymbol{\lambda}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \boldsymbol{\lambda} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \boldsymbol{\lambda} & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \boldsymbol{\lambda} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \boldsymbol{\lambda} \end{bmatrix} = \lambda^2 det\begin{bmatrix} \boxed{\boldsymbol{\lambda}} & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & \boldsymbol{\lambda} & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{\lambda} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{\lambda} & 0 & \mathbf{1} \\ 0 & \mathbf{1} & \mathbf{1} & 0 & \boldsymbol{\lambda} & 0 \\ 0 & 0 & 0 & 0 & \mathbf{1} & \boldsymbol{\lambda} \end{bmatrix}$$

$$= \lambda^3 det\begin{bmatrix} \boxed{\boldsymbol{\lambda}} & 0 & 0 & 0 & 0 \\ 0 & \boldsymbol{\lambda} & \mathbf{1} & 0 & 0 \\ 0 & 0 & \boldsymbol{\lambda} & 0 & \mathbf{1} \\ \mathbf{1} & \mathbf{1} & 0 & \boldsymbol{\lambda} & 0 \\ 0 & 0 & 0 & \mathbf{1} & \boldsymbol{\lambda} \end{bmatrix} = \lambda^4 det\begin{bmatrix} \boldsymbol{\lambda} & \mathbf{1} & 0 & 0 \\ 0 & \boldsymbol{\lambda} & 0 & \mathbf{1} \\ \mathbf{1} & 0 & \boldsymbol{\lambda} & 0 \\ 0 & 0 & \mathbf{1} & \boldsymbol{\lambda} \end{bmatrix} = \lambda^4(\lambda^4 + 1) = 0, \quad (28)$$

In this case, the characteristic equation for $[A]$ is:

$$[A]^4([A]^4 + [I]) = [0], \quad (29)$$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

which indicates that the network has four transient states and a single cycle of length 4. Since $tr([T]) = 0$, there are no fixed points. Higher powers of $[T]$ (computed via GF(2) operations, and having $\vec{M}_3$ in (5) as a row basis and a column basis) include:

$$[T]^2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \end{bmatrix}, \quad [T]^4 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \end{bmatrix},$$

$$[T]^3 = [T]^7 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 \\ 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & \mathbf{1} \\ \mathbf{1} & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & 0 \end{bmatrix}. \tag{30}$$

Since $tr([T]^2) = 0$, there are no loops whose length is a divisor of 2. The fact that $tr([T]^4) = 4$, means that there are four states on cycles of lengths equal to a divisor of 4. These states are states $x_3,\ x_1x_3,\ x_2x_3,\ and\ x_1x_2x_3$, or equivalently $001,\ 101,\ 011,\ 111$ since they correspond to the four diagonal elements of value 1 in $[T]^4$. Since there are no cycles of length 1 or 2, these four states form a single cycle of length 4. Equation (30) is the minimal relation of the form $[T]^{r_0} = [T]^{r_0+u}$ equating two powers of $[T]$. It asserts that the longest transient chain is of length 3, and that the least common multiple of all cycle lengths is 4. It also implies that

$$x_i(t+7) = x_i(t+3),\ \ i = 1, 2, 3. \tag{31}$$

At least one of (but not necessarily every one of) the three relations (31) is a minimal reduced scalar equation for the pertinent variable (Rushdi, 2015). In fact, the reduced scalar equations are not the same in this particular example, but they are:

$$x_1(t+6) = x_1(t+2),\ \ x_2(t+5) = x_2(t+1),\ \ x_3(t+7) = x_3(t+3). \tag{32}$$

Note that relations (32) imply (31), but the converse is not true. Now, we note that the characteristic equation (29) for $[A]$ is not the minimal one, since the total number of transient states $(four)$ exceeds the length of the longest transient state $(three)$. In fact, the minimal equation for $[A]$ should be (by virtue of (19) and (30)):

$$[A]^3([A]^4 + [I]) = [0]. \tag{33}$$

To finalize this example, we indicate that all our findings can be easily verified. Figure 2 shows a Karnaugh-map representation for the three next-state functions $f_1, f_2, and\ f_3$. For convenience, the current value of each map cell is shown in a small corner within the cell. From Figure 2, we can directly draw the complete state-diagram of the network in Figure 3. To check the number of

International Journal of Mathematical, Engineering and Management Sciences
Vol. 6, No. 2, 598-610, 2021
https://doi.org/10.33889/IJMEMS.2021.6.2.036

network cycles, we note that:

$$rank\,([T]-[I]) = rank \begin{bmatrix} \mathbf{1} & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{1} & 0 & \mathbf{1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathbf{1} & 0 & \mathbf{1} \\ 0 & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 & \mathbf{1} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{1} & \mathbf{1} \end{bmatrix}, \tag{34}$$

is obviously 7 (the sum of all rows is 0, and there is no other linear dependency). This means that the number of cycles is $(8-7)=1$.

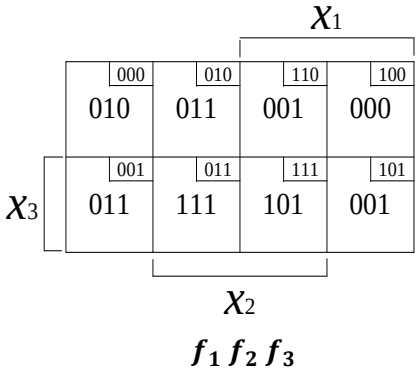

Figure 2. A Karnaugh-map representation for the next-state functions for the example network

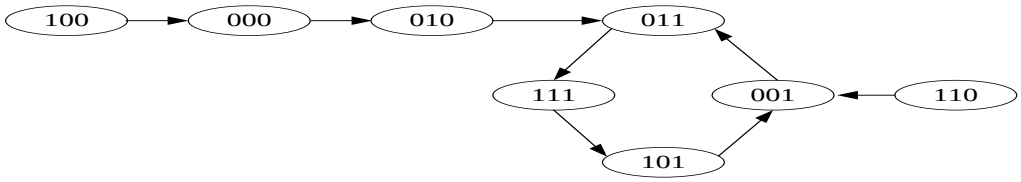

Figure 3. The state diagram of Example 1 with states designated by the binary values of the 3-tuple $\{x_1x_2x_3\}$. This is a complete recipe of all possible scenarios of transitions among network states. In any particular scenario, the network traverses only states on a specific transient path followed by a cycle.

4. Discussion

Starting from the scalar next-state functions of the network, we write its function matrix $[A]$ in terms of the various products of these functions. We then obtain the transition matrix $[T]$ utilizing its similarity with $[A]$, i.e., by pre- and post-multiplying $[A]$ with the state matrix $[S_n]$. This step is self-checking, since it must produce a matrix $[T]$ whose column vectors are binary unit vectors. Since the implicit basis vector for the column vectors of $[T]$ represents the *current states*, while that of its row vectors represents the *next state*, the matrix $[T]$ can be used to automatically draw the full state diagram of the network. However, we have chosen in our example to utilize

properties of $[T]$ to make direct deductions about network behavior without drawing the state diagram. We utilized the powers of $[T]$, or alternatively used its characteristic equation. The predictions made by the various methods are self-consistent and are in full agreement with results obtained earlier, notably via the STP machinery. These predictions also coincide exactly with those deduced from the full state diagram.

5. Conclusions

We utilized well-known results in the seminal work of Cull (1971), together with the improvement added by Rushdi and Al- Otaibi (2007), to present a method for constructing the transition matrix $[T]$ of an SBN. We then combined several approaches for inferring network behavior from mathematical properties of $[T]$ without drawing the full state diagram. Our work leads to a better understanding of the relation between the matrix and scalar approaches for studying SBNs. While our matrix method is *algorithmic* in nature and *complete* in scope, the scalar techniques use ad-hoc cumbersome heuristics, and might fail to predict the exact network behavior, especially its transient one. This paper is a continuation of earlier matrix methods, but it distinguishes itself by formulating a well-formed algorithm for constructing the matrix $[T]$, and by amalgamating various techniques for predicting network behavior from the mathematical properties of $[T]$. We attempt to make the matrix analysis of SBNs independent of any particular implementation, as we deliberately avoid the use of semi-tensor products (STP) as our matrix approach. Many papers utilize STP as if it were the only tool for formulating matrix equations for Boolean networks. Admittedly, the STP is a powerful computational tool, but it is inefficient and not easy to follow. Further comparison between our matrix method and the STP one is warranted.

Conflict of Interest

The authors assert that no conflict of interest exists.

Acknowledgments

This work is funded by the Deanship of Scientific Research (DSR), King Abdulaziz University (KAU), Jeddah, Saudi Arabia. Therefore, the authors acknowledge, with thanks, the DSR for their financial and technical support. The first-named author (AMAR) is gratefully indebted to Dr. Rufaidah Rushdi, of Kasr Al-Ainy Faculty of Medicine (Cairo University, Arab Republic of Egypt) for stimulating discussions concerning cellular biological networks.

References

Cheng, D. (2019). *From Dimension-Free Matrix Theory to Cross-Dimensional Dynamic Systems*. Academic Press, New York.

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

Cull, P. (1971). Linear analysis of switching nets. *Kybernetik*, *8*(1), 31-39.

Li, R., & Chu, T. (2012). Complete synchronization of Boolean networks. *IEEE Transactions on Neural Networks and Learning Systems*, *23*(5), 840-846.

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

Rushdi, A.M., & Al-Otaibi, S.O. (2007). On the linear analysis of synchronous switching networks. *Journal of King Abdulaziz University: Engineering Sciences*, *18*(2), 43-72.

Rushdi, A.M., & Rushdi, M.A. (2017). Switching-Algebraic Analysis of System Reliability. Chapter 6 in Ram, M. and Davim, P. (Editors), *Advances in Reliability and System Engineering*, Management and Industrial Engineering Series, Springer International Publishing, Cham, Switzerland, 139-161.

Rushdi, A.M.A. (2015). Derivation of reduced scalar equations for synchronous Boolean networks. *Journal of King Abdulaziz University: Computers and Information Technology*, *4*(2), 39-68.

Rushdi, A.M.A. (2018). Utilization of Karnaugh maps in multi-value qualitative comparative analysis. *International Journal of Mathematical, Engineering and Management Sciences*, *3*(1), 28-46.

Rushdi, A.M.A., & Ghaleb, F.A.M. (2016). A tutorial exposition of semi-tensor products of matrices with a stress on their representation of Boolean functions. *Journal of King Abdulaziz University: Computers and Information Technology*, *5*(1), 3-30.

Rushdi, R.A., & Rushdi, A.M. (2018). Karnaugh-map utility in medical studies: the case of Fetal Malnutrition. *International Journal of Mathematical, Engineering and Management Sciences*, *3*(3), 220-244.